An Interop Publication

ConneXions®

The Interoperability Report

September 1992 | *Special Issue: Electronic Mail and Directory Service* | Volume 6, No. 9

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779. Toll-free: 1-800-INTEROP. E-mail: connexions@interop.com.

ISSN 0894–5926

From the Editor

Electronic mail (e-mail) is certainly the most widely used of all network applications. Every day, thousands upon thousands of messages flow across public and private networks all over the world, enabling users to "get their work done." In this *Special Issue,* we will look at some aspects of e-mail (both Internet and OSI) and directory services. Directory systems such as X.500 are envisioned to play an important role in the world-wide deployment of e-mail, thus it is natural to look at these two technologies in conjunction.

We start with an overview of the 1992 extensions to X.500. The 1988 version of the X.500 standards lacks a number of critical pieces of functionality required to build truly interoperable, distributed, enterprise-wide Directory services. The 1992 study period, currently coming to a close, has focused entirely on addressing these missing pieces of functionality by defining a set of "Extensions" to the 1988 X.500 standards. Sara Radicati outlines the 1992 extensions work and examines what impact it might have on product development.

While standardization work is an important continuing effort, implementation and deployment of new technologies also play a vital part on the road to fully operational OSI networks. In two articles, Steve Hardcastle-Kille outlines the *QUIPU* Directory implementation, and the PP *Message Transfer Agent* (MTA).

X.400 can also be used for so called "mail-enabled applications," and one example is *Electronic Data Interchange* (EDI). EDI provides an automated way for businesses to exchange structured business data such as invoices, purchase orders, and other financial documents. Sara Radicati explains how EDI can be made to use X.400 as the underlying transport and infrastructure mechanism.

Because of its relative low cost and ability to "run over anything," e-mail is an important enabling technology that let's you "get started" in networking. Our user perspective comes from Zambia and is written by Mark Bennett.

The *Multi-Media Internet Mail Extensions* (MIME) standard was recently developed to adapt basic RFC 822 messaging to operate in a brave new world which includes non-ASCII text and multi-media attachments. We asked Greg Vaudreuil to give us an overview of MIME. This is followed by a comparison of X.400 and Internet mail technologies, written by Daniel Blum.

One of the reasons e-mail has been so successful is that you can reach people on different networks using different technologies. This is accomplished through the use of *mail gateways.* Marshall Rose explains the theory and practice of mail gateways.

The 1992 Extensions to X.500

by Sara Radicati, Creative Network Solutions

Background

In an effort to bring the 1988 CCITT study period to a close within the prescribed four year CCITT study cycle, the designers of 1988 X.500 *Directory Service* were forced to make some hard decisions about the scope of the first round of recommendations. The result was that the 1988 version of the X.500 standards lacks a number of critical pieces of functionality required to build truly interoperable, distributed, enterprise-wide Directory services. In particular, it does not yet provide standardized support for access controls or a full set of replication protocols to support the automatic synchronization of remote replicas of the database. Vendors developing Directory systems based on 1988 X.500 today are often doing so by defining proprietary mechanisms to support access controls and replication.

The 1992 study period currently coming to a close, has focused entirely on addressing these missing pieces of functionality by defining a set of "Extensions" to the 1988 X.500 standards. This article describes in detail the 1992 Extensions work and examines what impact it is likely to have on product development.

What is missing from 1988 X.500

The major items which dropped out of the 1988 X.500 standards were: *Replication, Access Controls, Schema* and *Knowledge Management.* The lack of each of these features has had a different impact on the development of 1988 X.500 products as well as on the current deployment of X.500 services. In particular, the lack of standardized Access Controls and Replication have contributed to slow down the commercial deployment of X.500 services.

Replication

While the 1998 standards permit the existence of copies of *Directory Information Tree* (DIT) information, they provide no automatic support for updating information once it has been replicated. The lack of Replication protocols in 1988 X.500 meant that vendors building distributed directory services, such as DEC and HP, have had to "roll" their own replication protocols. In fact, most of the distributed enterprise directories being released on the market today implement the standard DAP and DSP protocols as mandated by 1988 X.500, along with a set of proprietary extensions necessary to support replication.

This is clearly less than was originally desired when the X.500 work began. However, it is also true that in most cases today there is, as yet, little need for Directory database servers (DSAs) from different manufacturers to exchange data. This makes the current situation where replication protocols among different vendors are incompatible tolerable for the immediate period. Nevertheless, as enterprise directories grow, it will become increasingly necessary in large distributed environments to have compatible replication protocols.

Access Control

In the case of Access Controls, the 1988 situation is simply embarrassing. The 1988 standards define a complex and articulate framework for authentication (X.509) which has formed the basis for work on security and authentication in both ISO and CCITT, while at the same time leaving the issue of access controls completely unresolved. Unfortunately, authentication without access controls is of little value—e.g., it matters little that "Joe Smith" is identified through the use of authentication to be truly Joe Smith if there isn't also a standardized mechanism of deciding whether Joe Smith has the right to read the corporate personnel files or not.

Defining authentication mechanisms without also standardizing on a set of access control policies and mechanisms has left the issue of security in X.500 Directories wide open. Needless to say, security is of paramount concern for most corporate users and government agencies wishing to implement distributed directory services.

Vendors releasing X.500-based products have dealt with this situation by implementing simple forms of proprietary access control mechanisms to go along with their implementation of the X.500 standard. Again this results in products from one vendor which are incompatible with products by other vendors. Lack of standardized access controls makes interoperability among products from different vendors even more difficult than the lack of replication protocols. The only way that a DUA (*Directory User Agent*) from one manufacturer today can successfully query a DSA (*Directory Service Agent*) from a different manufacturer is if the access controls parameters are turned off. Clearly this is an impractical scenario in most production environments.

Knowledge Management

Knowledge Management is closely tied to replication and refers to the ability to manage the information which the Directory service holds internally to help it keep track of how information has been partitioned among multiple Directory servers. This internal directory information is called "knowledge" and is used internally by the directory to locate information in response to user queries. Clearly as directory servers (DSA) are added or removed from a network, it would be highly desirable for the directory servers in the network to automatically update their knowledge information to reflect the new disposition of the information database. With the 1988 version of the standard these changes must be tracked and executed manually. This means that in 1988 implementations of X.500, system administrators are typically required to track changes to the knowledge information, and rely on off-line mechanisms to update each DSA's internal knowledge references. Interoperability between products is not as heavily affected as in the case of replication or access controls, as each product comes with its own configuration mechanisms and while it is a considerable nuisance for system administrators to rely on different tools and procedures for different products, it does not impact the operational behavior of the distributed directory environment as far as the user is concerned.

Some less significant items also dropped out of 1988 X.500, such as distributed entries and operations acting on multiple entries. Distributed entries are entries made up of information which resides in different database servers. The need for such entries is an issue dear to many public service providers (PTTs) which envision, potentially, the need to distribute large amounts of information describing national services and computing resources across multiple directory servers. While the 1992 study period made some attempts to deal with the need for this feature, it has eventually ended up postponing the design of a mechanism to support distributed entries to the 1996 study period.

The need for operations acting on multiple entries, on the other hand, was not revisited at all in 1992 as it was generally felt that for the most part the same functionality could be achieved through the use of the Search operation as originally defined in 1988 X.500.

The 1992 extensions

The 1992 Extensions to X.500 have focused primarily on addressing three areas not covered by the 1988 standards for Directory services: Access Controls, Replication and Schemas.

continued on next page

The 1992 Extensions to X.500 *(continued)*

However, in addressing these three major areas of functionality the designers of 1992 X.500 realized early on that there was a common thread among them in that the information model defined in 1988 X.500 was not sufficient to accurately model some of the functionality required to describe the internal behavior of the directory service.

This realization led early on to the development of what has been called the *Extended Information Model,* which provides the mechanisms necessary to more clearly describe the internal behavior of the directory.

The development of the new information model has had a pervasive effect throughout all parts of the X.500 standard, and has resulted in Addenda (i.e., new text) to nearly all of the documents which comprise the original 1988 standard. Fortunately, for the most part these changes are of an additive nature and will not have an adverse impact on migration from the 1988 to the 1992 versions of the standard.

Views

The 1992 Extended Information Model forms the basis for the design of Access Controls, Replication and Schema in 1992 X.500. It specifies that there are two distinct "views," or models of Directory information:

- *The User Information Model:* which corresponds to the attributes and object classes as were originally defined in the 1988 version of the standard, and
- An *Operational and Administrative Information Model:* which defines a new set of facilities necessary for modelling 1992 information.

The intent of the Operational and Administrative Information Model is that it can be overlayed over the 1988 User Information Model without having to replace or obsolete it. This means that 1988 protocol operations should still work when directed at 1992 directory servers (DSAs), and vice versa 1992 directory user agents (DUAs) will be able to query 1988 directory servers (DSAs). The difference will lie in the fact that 1992 protocols will also be able to query and modify administrative information not available in 1988 implementations.

The 1992 Information Model relies on two new concepts in order to model information which is specific to the directory service's own internal use:

- *Operational Attributes,* and
- *Subtrees.*

Operational Attributes

Operational Attributes are similar in structure to regular attributes but are intended to be defined for use by the Directory service only. That is, they are not visible through ordinary user queries. It is envisioned that in 1992 X.500 Directory entries will have both user attributes (as were defined in 1988 X.500) and operational attributes as defined in 1992. Operational attributes will be used to store information such as access control permissions, knowledge information (i.e. the information which directory servers rely upon to locate data in the distributed directory), and schema information which defines the allowable superior/subordinate relationships of entries in the directory.

1992 Directory entries will therefore comprise: 1988 naming and user attributes, and may in addition contain any number of operational attributes necessary to store information required for the directory's own internal operation, such as the access controls which pertain to that particular entry, or the name and address of the directory server (DSA) which holds the master copy of the entry.

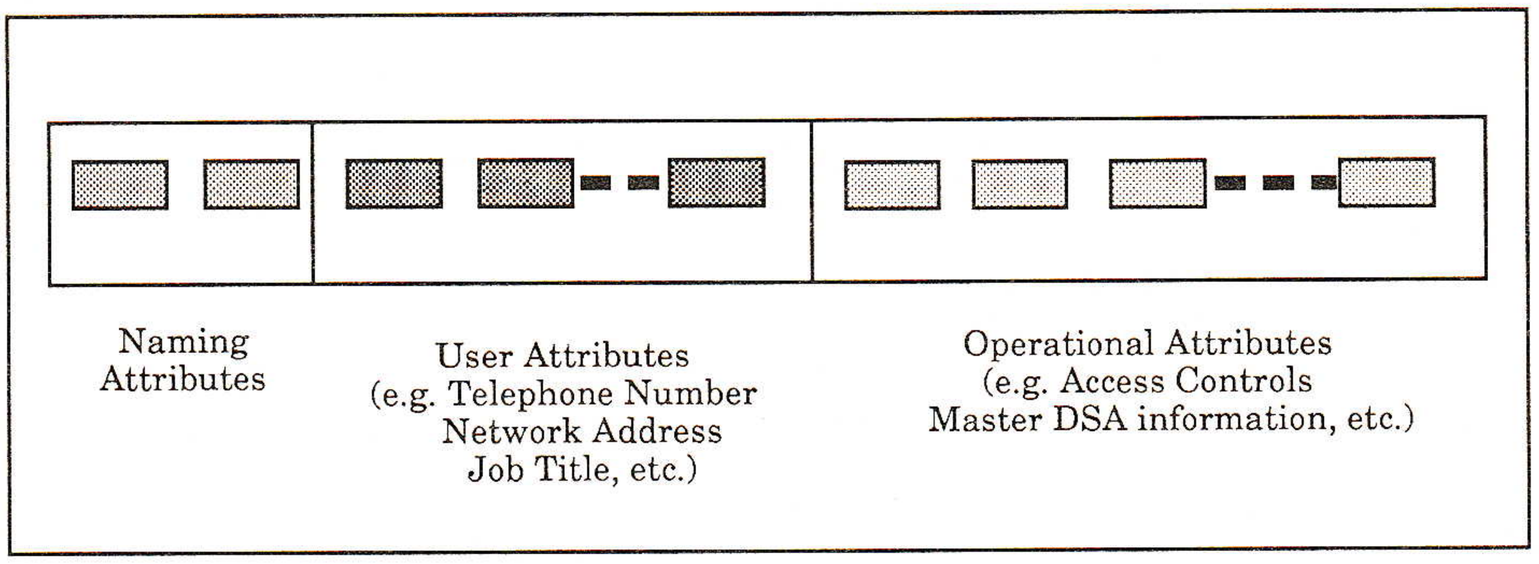

Figure 1: 1992 Directory Entry Structure

Operational attributes can be associated with single entries, individual attributes, attribute values, or collections of entries also called subtrees. Operational attributes are further specialized into 3 categories:

- *Directory operational attributes:* which may be used to model features such as Access Controls.
- *DSA-shared operational attributes:* which are used to model the information which DSAs will need to make available to other DSAs in order to carry out replication functions.
- *DSA-specific operational attributes:* which serve to describe operational information that is specific only to a particular DSA implementation, such as the last time it received an update from the master DSA.

Subtrees

Subtrees are another very powerful concept in 1992 X.500. Subtrees serve to denote dynamic subsets of DIT information. They may overlap, and span DSA boundaries. In particular, they can be defined dynamically by specifying an initial entry, a set of boundary entries, and a set of filtering criteria which can be used to include or exclude entries on the basis of information they contain. The use of subtrees and operational attributes in 1992 X.500, makes it possible for example to associate access controls with entire sub-portions of the Directory information base.

Figure 2, on the next page, illustrates the use of subtrees and operational attributes. The entry "XY," for example, may contain a subtree definition which encompasses all the entries within the dotted frame. The subtree definition will be represented within entry XY as a set of operational attributes which only the directory service is aware of.

A set of access controls can also be stored at entry XY again as a set of operational attributes which will have an effect on the entire subtree defined at XY. Since the 1992 operational attributes can be manipulated just like any other attribute, the scope of the access controls stored at XY or their semantics can be easily changed through protocol operations.

continued on next page

The 1992 Extensions to X.500 *(continued)*

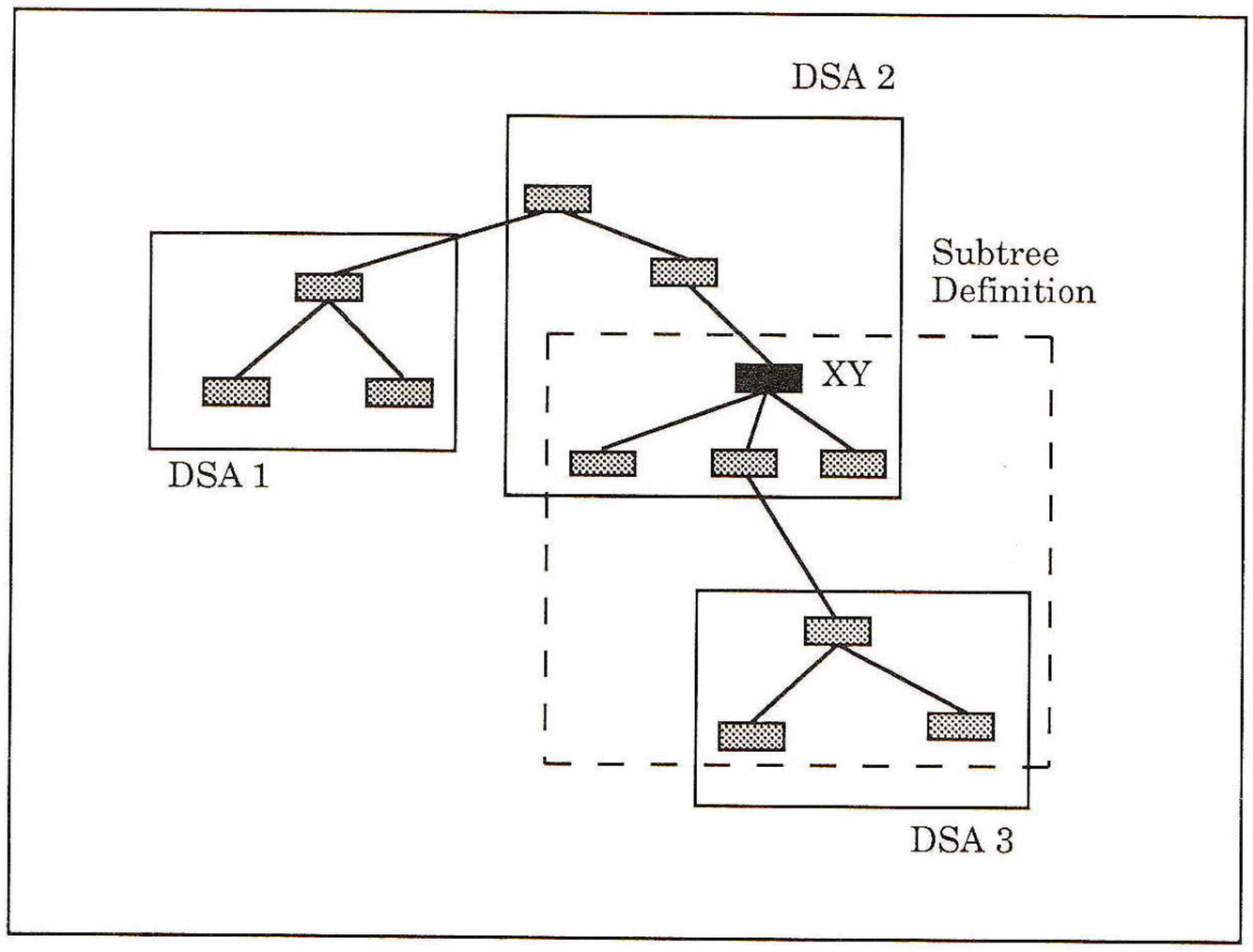

Figure 2: Subtrees and Operational Attributes

Access Controls

While authentication serves to verify a user's identity and may also be used to verify the origin of information received, Access Controls are used to restrict access to information on the basis of identity. The 1992 extensions work defines a set of mechanisms to support Access Controls based on access control lists. 1992 X.500 provides a mechanism whereby different access control schemes can be dynamically "bound in" provided both cooperating parties can agree on which access control scheme to use. This means that future access control schemes can be defined and used in conjunction with the 1992 protocols. It also means that current proprietary access control schemes can continue to be used in conjunction with the new 1992 protocols, thus easing product migration forward to 1992. Work on Access Controls is expected to resume again in the 1996 study period, in order to define additional forms of access controls based on more sophisticated mechanisms such as capabilities.

The 1992 list-based access control scheme is called "basic-access-control." The basic access control scheme may be used to protect entries, or operational attribute information including access control information itself. Items which may be protected via the basic-access-control scheme include: entries, attributes or attribute values. Access Control information stored as operational attributes may be associated with entries, attributes, or with entire subtrees so as to affect an entire portion of the Directory information base. User classes may be associated with the access control permissions to denote groups of users which may or may not have access to particular pieces of information.

Replication mechanism

1992 X.500 relies on a strict Master/Shadow model of replication. That means that updates can be directed only to the Master of the information. However, indirect shadows, that is shadows of shadows are allowed. A DSA may be a Master for some portions of information while being a Shadow DSA for other portions of information.

Replication in 1992 X.500 relies heavily on the use of operational attributes. Operational attributes are used to represent knowledge information, such as superior and subordinate references, as well as master/shadow relationship information.

Replication in the 1992 version of the standards caters to two basic requirements of Directories: improved availability of information, as well as increased performance by ensuring that data which is often accessed is located close to the DUAs making the request. The 1992 version of the standards will support two forms of replication: *caching,* and *shadowing* of information. Caching is already available in 1988 Directories and allows DSAs to keep copies of information, but provides no guarantee of being up to date. Shadowing, on the other hand, involves a contract between two DSAs whereby a Master DSA ensures that Shadow DSAs receive updated copies of information on an agreed upon schedule.

The 1992 standards continue to adhere to the principle of "transient inconsistency" in that they do not require all shadow copies of information to be updated simultaneously. Updates can only be directed to the Master of the information which is then responsible for updating the shadow copies. Access Control information is replicated along with the information it protects to ensure consistency of service throughout the DIT.

Replication protocols

Two new protocols are defined to handle Replication in the 1992 Extensions:

- *The Directory Information Shadowing Protocol* (DISP), and
- *The Directory Operational Binding Management Protocol* (DOP).

The DOP is used between any two DSAs which are entering into an association agreement either to shadow each other's database information or to keep reference pointers (i.e., knowledge information) to each other up to date. Two DSAs must first enter into an operational binding agreement through the use of the DOP protocol, before the DISP protocol can be used to replicate and update information among them. The DOP protocol allows DSAs to negotiate the nature of the binding agreement and any parameters which will govern the association, such as the frequency with which update information will be sent from the Master DSA to the Shadow DSA.

The DISP protocol is used between shadowing DSAs to actually transfer information across from one DSA to another, as well as transmit any updates. Since the DISP protocol is expected to have to transfer potentially very large amounts of information among DSAs it was felt that it should support a reliable bulk data transfer mechanism to ensure the correct delivery of the information. The RTSE protocol already used in X.400 to support MTA to MTA transfers was chosen to provide the optional underlying bulk-data transfer facility for implementing the DISP functionality.

Lack of Knowledge Update protocols

The 1992 Replication work focused almost entirely on the shadowing aspects of replication and only begins to address the requirements for updating Knowledge information through the provision of the DOP protocol. Clearly any two DSAs holding knowledge references which point to each other (or from one to the other) must be engaged in an operational binding agreement. However, the existence of an operational binding agreement alone does not ensure that changes in knowledge information are automatically propagated to all the DSAs affected by the change. It is expected that the topic of knowledge management will be addressed again by the 1996 standardization work.

Schema

1988 X.500 introduced the concept of *Schema,* that is, rules that defined the permitted relationships between entries of different object classes.

continued on next page

The 1992 Extensions to X.500 *(continued)*

The overall intent behind schema rules was to avoid inappropriate structuring of directory information such as placing the name of a country underneath the entry for a person's name. The idea was to have agreed-upon structure rules that the entire Directory database would abide by, thus making it easier to search for particular types of information. In 1988 X.500, however, schema are only a concept, and while examples are given, no hard and fast mechanisms are ever provided to describe schema relationships and therefore define how schema rules are to be enforced.

In 1992, the concept of operational attributes is again used to represent schema information. Finally, through the use of operational attributes, schema rules can be represented as attributes contained within directory entries and can therefore be read and modified to reflect the structure of the information base. By simply being able to store schema definitions within the Directory as operational attributes it becomes possible to rely on schema definitions to control the modification of the directory information base and ensure that a consistent structure is maintained.

Migration from 1988 to 1992 X.500

The best news about the 1992 extensions to X.500 is that for the most part they are fully backward compatible with 1988 X.500. The new Information Model augments the 1988 Information model rather than altering it substantially. Best of all, it will still be possible to use 1988 DAP protocols to query a 1992 Directory, and vice versa it will be possible to use 1992 DAP protocols to query 1988 Directories.

The interesting question, however, becomes to what extent will vendors discard the proprietary extensions they have developed for 1988 X.500 in order to adopt the 1992 extensions into their products. This is of particular interest where key features such as Access Controls and Replication are concerned as these ultimately affect interoperability between products from different manufacturers.

We expect that 1992 Access Controls and Replication extensions will find their way differently into products. Since Access Controls are visible to the DAP protocol user, we expect that vendors will be highly motivated to adopt the new 1992 Access Control scheme as soon as possible within their products. One of the primary goals of vendors of Directory services is to allow their DSA databases to be queried from any vendor's DUA.

The fact that the 1992 Access Control scheme is designed to allow the dynamic binding of any access control mechanism, will make it easier for vendors to develop a smooth transition plan for phasing out their proprietary 1988 Access Control mechanisms in favor of 1992 Access Controls. We expect, however, that the proprietary access control schemes will continue to exist alongside the newly defined standard access controls for a long while to come.

The 1992 Replication extensions, on the other hand, are likely to find their way much more slowly into vendor's products. Similarly to Access Controls, the 1992 Replication protocols can be implemented alongside the 1988 DSP protocol and therefore do not present a major migration problem. Figure 3 shows how the 1988 protocol structure of DAP and DSP can essentially remain in place while being augmented with the 1992 protocols to provide replication.Vendors wishing to incorporate the new 1992 Replication protocols into their architectures do not need to replace the 1988 DSP protocols entirely, but can choose to simply run the new DOP and DISP protocols alongside a 1988 DSP.

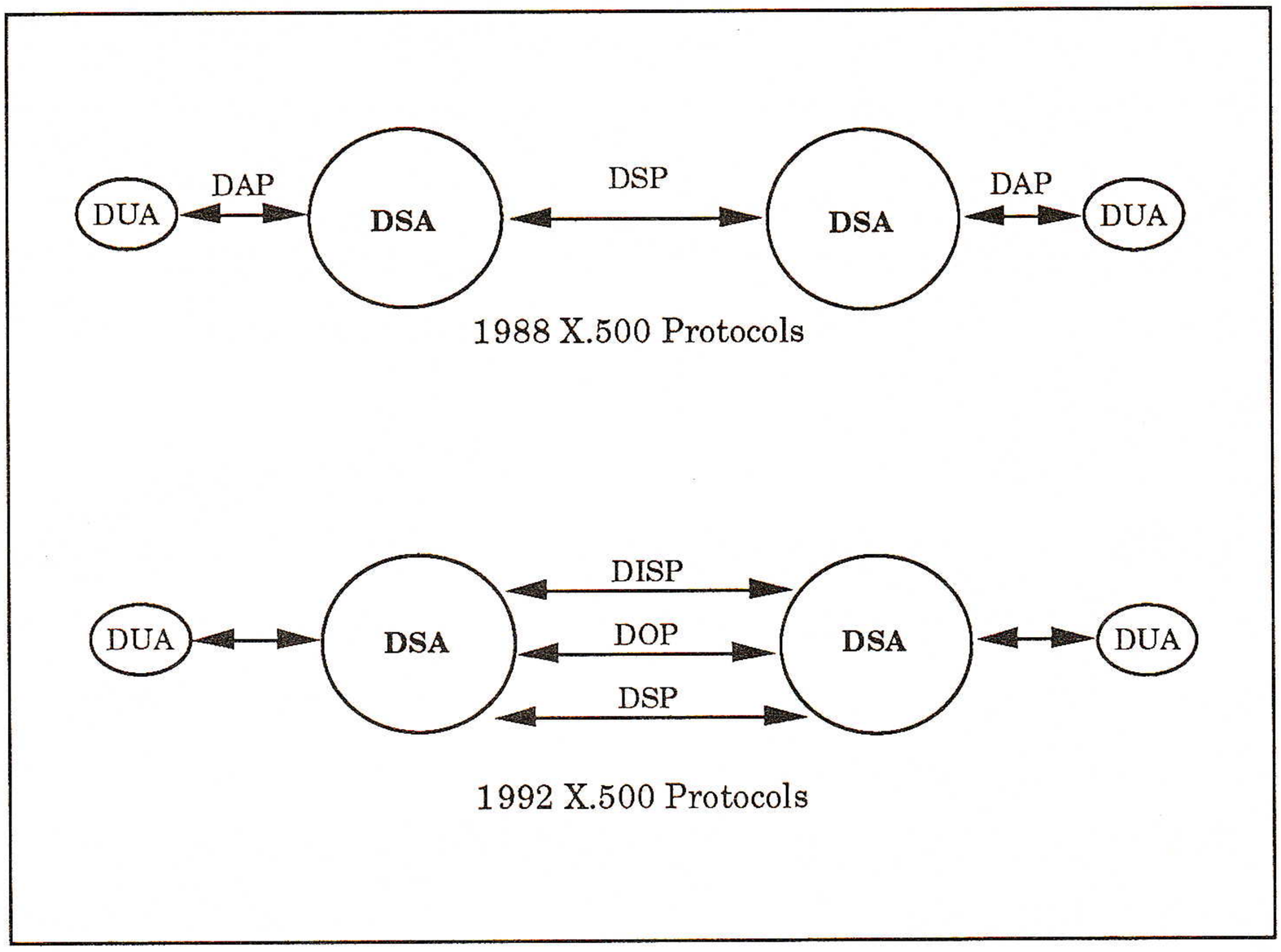

Figure 3: 1988 versus 1992 Protocol Migration

However, most vendors have already implemented proprietary versions of the DISP and DOP protocol. Since the pressure for interoperability between DSAs from different manufacturers is significantly smaller than that for DUA to DSA interoperability, it is less likely that vendors will rush to throw away their proprietary replication protocols in favor of the DOP and the DISP. In addition, both the DOP and DISP protocols are as yet wholly untested pieces of technology. This means that developing fully interoperable versions out of the 1992 protocol specifications will take several years of testing and experimentation, before yielding implementations which are at a level adequate for commercial deployment.

We expect vendors to be very cautious about replacing their proprietary replication protocols with the 1992 Replication protocols. Unfortunately, this means that interoperability between DSAs from different manufacturers will continue to be impeded.

References

[1] Benford, S., "Components of OSI: X.500 Directory Services," *ConneXions,* Volume 3, No. 6, June 1989.

[2] "NYSERNet Sponsors White Pages Pilot," *ConneXions,* Volume 3, No. 9, September 1989.

[3] Marshall T. Rose, *The Little Black Book: Mail Bonding with OSI Directory Services,* Prentice Hall, Inc., ISBN 0-13-683210-5, 1992.

[Reprinted with permission from *The Messaging Technology Report,* a monthly research report focusing on the latest trends in the Message Handling and Directory Services industry. The report is published by Creative Network Solutions. Phone: 415-857-0963]

SARA RADICATI is president of Creative Network Solutions. She has been involved with the development of electronic mail technologies and standards for the past ten years. She has been a principal contributor to the design of the X.500 standards for Directory Services and she has participated in major standardization forums, including ISO, CCITT, ANSI and NIST. Dr. Radicati was most recently responsible for Messaging Products and Strategies at Novell, Inc. Prior to joining Novell, Dr. Radicati was responsible for product architecture at Xerox Corp. She holds M. S. and Doctor of Science degrees from the California Institute of Technology and the University of Pisa, Italy.

The QUIPU Directory Implementation

by S. E. Hardcastle-Kille

Introduction

This article describes the *QUIPU* Directory Implementation, which is an implementation of the CCITT X.500 Protocols [3]. (The reference here is to the existing 1988 X.500 specifications, and these are assumed by default. The 1992 recommendations, which are expected to be available soon, are referenced explicitly). The X.500 Directory Service is a highly distributed database, which provides a rich and extensible information framework and a range of services to read, modify, and search this information. Objects are named globally, using a hierarchical scheme. For example:

```
Common Name=Ole Jacobsen,
Organisation=Interop Company,
State=California,
Country=US
```

Components of names are typed to facilitate naming a very wide range of objects, but this information will usually be hidden from the end user. Information will typically be located by a series of search operations based on user input. For example, the name above might be located from the input:

```
O. Jacobsen, Interop, CA, US
```

The OSI directory has potential to provide a wide range of services, including:

- White pages services on people and organisations, to look up a wide range of information.
- Support for applications, such as message handling, to deal with addressing and configuration information.
- Basic infrastructure to support public key based authentication [2].
- Support for advanced distributed information services, such as dealing with bibliographic information.

There is broad acceptance in an increasing number of communities that X.500 based directory service will be a key component of future infrastructure. This is reflected in an increasing level of pilot activity, which is discussed later. This article describes QUIPU, which is the most widely deployed implementation of X.500.

History

The first version of QUIPU was developed in 1986 to facilitate experimentation with the OSI directory prior to the provision of large scale services. It was developed at University College London under the INCA (Integrated Network Communication Architecture) project. The Incas of Peru did not have writing, but they did store information on property and taxes on devices known as "Quipus." A Quipu is a set of strings, carefully knotted in a specific manner and with coloured thread, and attached to a larger rope. Thus, the QUIPU is the INCA directory service.

The early QUIPU prototype was successful, and was used to give the first demonstration of distributed X.500 at the ESPRIT Conference Week in November 1988. This demonstration involved Directory System Agents (DSAs) in four countries.

The viability of the system as being a suitable basis for operating a pilot service was demonstrated, and further development was undertaken to make this possible. This work was done in the UK and the US, with assistance from many people around the world involved with piloting activities. A key step was to integrate QUIPU into the ISODE Package of OSI Protocols, and to make it openly available to all interested parties [13].

The QUIPU DSA

The information stored in a directory service is known as the *Directory Information Tree* (DIT). A core component of an X.500 Directory Service is a *Directory System Agent* (DSA), which is a server that holds a part of the DIT. DSAs operate collaboratively, so that when a user accesses a DSA, information on any part of the DIT may be returned, irrespective of whether the DSA accessed holds the requested information. It is this collaboration to provide the distributed directory, and in particular distributed searching, that is the key feature of X.500.

The two main components of a DSA are the database which stores the DIT information, and the protocol engine which allows the DSA to communicate with other DSAs and *Directory User Agents* (DUAs). The QUIPU DSA holds its data in main memory, which it reads in from disk at startup. The disk format is text-based and structured in a manner which is easy to manipulate. The memory representation is aligned closely to the data structures of X.500. This allows the directory operations to be implemented in a very natural and general manner. In practice, not all data is resident in memory, and much is "paged out" to disk by the operating system. These basic structures are supplemented by indexes to improve the performance of access to identified entries and of common search operations. Because the DSA takes a significant time to start, it is implemented as a static process. This in turn requires that it performs scheduling to handle multiple simultaneous queries. This scheduling, together with the handling of distributed operations, is the glue between the database and protocol engine.

Protocol support

X.500 specifies two protocols:

- *Directory Access Protocol* (DAP) is for a DUA to access a DSA.
- *Directory Systems Protocol* (DSP) is for a DSA to access another DSA.

QUIPU fully implements both of these protocols and the associated services for reading, modifying and searching the Directory. The QUIPU protocol engines are generated using ASN.1 compilers, and interfaced onto the OSI Stack provided by ISODE. Use of automated techniques makes the protocol aspects straightforward to handle.

Internet extensions

The QUIPU implementation made a number of extensions to X.500. Initial work with deploying QUIPU showed that some of these extensions were fundamental to deploying an effective service. The *Internet Engineering Task Force* (IETF) established a working group on OSI Directory Services, in order to specify what is needed to deploy a service on the Internet. It is important that the operation of a service does not depend on implementation specific function, and so a number of extensions to X.500, largely derived from the QUIPU experience, have been specified for use on the Internet.

continued on next page

The QUIPU Directory Implementation *(continued)*

These extensions are:

- *Schema:* X.500 defines a basic information framework, giving objects (e.g., people, organisations, and countries) and a basic selection of information (attributes) which can be associated with these objects. As anticipated by X.500, a wide range of additional information is needed to handle the wide range of things that real organisations need to place in the directory. A list of such things, which range from room numbers, to favourite drink, to photographs, are specified in the COSINE and Internet Schema [1].
- *Operation over TCP/IP:* To be effective in the research community, it is necessary to operate X.500 over non-OSI lower layers. In particular, the directory is operated over TCP/IP, according to RFC 1006 which specifies how to provide the OSI Transport Service over TCP/IP [12]. This is coupled with extensions to the procedures for distributed operations, so that some DSAs can operate over TCP/IP and others operate on a full OSI stack.
- *Replication:* To make the directory operate efficiently and robustly, it is necessary to have replication of information. This is particularly important for "top level" information, and for information about how the DIT is mapped onto DSAs (technically known as "knowledge"). A simple mechanism for replication, known as Internet DSP, was defined to provide a workable solution for replication prior to the completion of OSI Standards for replication and available implementations of these standards [6].

Access control

X.500 does not define any procedures for access control. Data that is placed in a directory is usually sensitive, and so it is often very important to control who has access to read, modify, and search information in the Directory. QUIPU defines two types of access control. The first refers to single entries, and gives control as to who can read and modify attributes within the entry. This might be used to protect sensitive attributes from being read generally, and to give the user rights to modify certain attributes, and the administrator different rights. These rights may be inherited to simplify administration. A similar, but more complex mechanism will be available in the 1992 X.500. The second QUIPU mechanism goes beyond the 1992 standard, and provides mechanisms to control listing and searching. This is very important, as most organisations do not wish to have their directory "trawled." However, prohibiting searching makes the directory almost useless, as the real power is to identify entries by searching when the name is not known. The QUIPU implementation provides mechanisms to allow the full power of searching, but prevent large amounts of data from being extracted.

Management

Management of a distributed directory is a complex operation, and a whole range of management tools have been developed around the QUIPU system. Tools produced include:

- Tools to facilitate installation and setup.
- Tools to monitor and probe the operating directory.
- Tools to change the operational configuration.
- Tools to "bulk load" data from external sources into the directory.
- Tools to manipulate data within the directory.

The DUA library

The QUIPU development was primarily concerned with the DSA and provision of directory service. QUIPU specifies a C API (Application Program Interface), to facilitate the implementation of *Directory User Agents* (DUAs). The QUIPU package has always contained a number of DUAs, some as demonstrations and some for serious use. The approach of providing the API has clearly been effective, as many more DUAs have been implemented by organisations not associated with the QUIPU development. Some of these are research prototypes, and others are commercial products. The QUIPU API has proven itself an appropriate and effective type of API for this function.

DUAs

There is not space in this article to give a full list of all the DUAs which are associated with QUIPU. In a survey of X.500 implementations, Lang and Wright identify fourteen DUAs associated with QUIPU [10], and there are several more known. Fuller descriptions of selected DUAs, together with illustrations are given in the second PARADISE international report [11]. Sample DUAs based on QUIPU are:

- *DISH* (the DIrectory SHell) which was implemented along with QUIPU is a useful tool for testing the directory and for management functions.
- The *FRED* DUA, implemented by PSI, is used in the PSI White Pages Pilot, which emulates the older Internet WHOIS service.
- The Apple Macintosh maX.500 implementation from the University of Michigan accesses QUIPU using the lightweight DIXIE protocol [7].
- The *XT-DUA* for the X Window System is a product commercially available from X-Tel Services.

Performance

The following figures are in the context of typical pilot usage, with DSAs running on medium sized workstation type servers. The memory oriented database attracted much initial criticism, but its performance has exceeded the expectations of its strongest proponents.

There is a cost of about 0.5Kbytes of memory for each entry, and virtual memory works well in conjunction with the database. Reads take a fraction of a second, and searches which make use of indexing take about a second. Searches which do not use indexing take about a second for every thousand entries searched. The major problem with scaling is not response time, but the DSA startup time. The practical limit for this technology seems to be around 100,000 entries in a DSA. As well as database speed, there are other factors on performance, including:

- Time to establish associations.
- The response time and throughput of the underlying network.
- The way in which distributed operations work.

In practice, the response is sometimes excellent and sometimes appalling.

Deployment

The deployment of QUIPU in directory pilot activities around the world has been a major success for QUIPU. Many DUAs have been used in these pilots, but the majority of DSAs are QUIPU. The first pilots were the PSI White Pages Pilot in the US [14] and the UK Academic Community Pilot [5].

The QUIPU Directory Implementation *(continued)*

These pilots, and others that emerged later have been brought together into a coherent international pilot by the PARADISE project. The current pilot operates in 23 countries, and contains over 700,000 entries on over 1700 organisations in over 300 DSAs. The primary use of this directory is for obtaining information on people, but experiments are starting with a range of other services, including document retrieval and support of message handling.

Future directions

Future releases of QUIPU will be made by the ISODE Consortium. There are many areas which it is planned to develop:

- *Database Work:* A new database API will be provided to allow multiple databases within a QUIPU DSA. This will include:
  - The current memory database
  - A new disk database designed for directory use
  - Sample mappings onto selected database products
  - Access to remote databases
- *1992 Extensions:* The 1992 X.500 recommendations will be supported.
- *Lightweight Protocols:* Lightweight protocols to access the DSA, suitable for implementation on smaller machines.
- *Multiple Provider:* A key problem is to allow multiple providers to participate in provision of directory services. This work, beyond 1992 X.500, is essential for commercial provision. This area is being examined by the North American Directory Forum.
- *Security:* Addition of X.509 (public key) based authentications [2].
- *Conformance:* Conformance testing, in addition to the extensive interoperability testing already undertaken.
- *SNMP Monitoring:* Monitoring of DSAs using SNMP [4].
- *Management:* Much work to improve and extend management features.

For further information

The QUIPU Manuals, together with a chapter on the architecture of QUIPU and a design analysis is packaged in a book [8]. The manuals may be obtained separately with the QUIPU Software [9]. For more information on the deployment of QUIPU in the international directory pilot, there is a brochure available from the PARADISE project, which gives useful information [11].

Getting the QUIPU software

The ISODE Package, which contains QUIPU, may be accessed by FTP from `cs.ucl.ac.uk`, `uu.psi.com`, or `archive.eu.net`. Retrieve the file `isode-7.tar.Z` in binary mode from the `isode/` directory. This file is the *tar* image after being run through the *compress* program and is approximately 6Mb in size. Future releases of QUIPU will be released by the ISODE Consortium [17].

References

[1] P. Barker and S. E. Hardcastle-Kille, "The COSINE and Internet X.500 schema," RFC 1274, November 1991.

[2] "The Directory—authentication framework," CCITT Recommendation X.509, December 1988.

[3] "The Directory—overview of concepts, models and services," CCITT X.500 Series Recommendations, December 1988.

[4] J. D. Case, M. S. Fedor, M. L. Schoffstall, and J. R. Davin. A Simple Network Management Protocol, RFC 1157, May 1990.

[5] J. A. I. Craigie, "UK academic community directory service pilot project," *Computer Networks and ISDN Systems,* 17:305–310, 1989.

[6] S. E. Hardcastle-Kille, "Replication and distributed operations extensions to provide an internet directory using X.500," RFC 1276, November 1991.

[7] T. Howes, M. Smith, and B. Beecher, "DIXIE Protocol Specification," RFC 1249, July 1991.

[8] S. E. Kille, *Implementing X.400 and X.500: The PP and QUIPU Systems;* Artech House, 1991, ISBN 0-89006-564-0.

[9] S. E. Kille and C. J. Robbins, "The ISO development environment: User's manual (version 7.0)," July 1991, Volume 5: QUIPU.

[10] R. Lang and R. Wright, "A catalog of available X.500 implementations, January 1992, (Internet Draft).

[11] "PARADISE international report (2)," November 1991. PARADISE Project Report.

[12] Marshall T. Rose and Dwight E. Cass, "ISO Transport Services on top of the TCP," RFC 1006, May 1987.

[13] M. T. Rose, "The ISO development environment: User's manual (Version 6.0)," January 1990.

[14] M. T. Rose, "Realizing the White Pages using the OSI Directory Service," Technical Report 90–05–10–1, Performance Systems International, Inc., May 1990.

[15] S. Benford, "Components of OSI: X.500 Directory Services," *ConneXions,* Volume 3, No. 6, June 1989.

[16] S. E. Hardcastle-Kille, "PP use of directory," Research Note RN/92/14, Department of Computer Science, University College London, January 1992.

[17] S. E. Hardcastle-Kille, "An Introduction to the ISODE Consortium," *ConneXions,* Volume 6, No. 5, May 1992.

[18] "NYSERNet Sponsors White Pages Pilot," *ConneXions,* Volume 3, No. 9, September 1989.

[19] Marshall T. Rose, *The Little Black Book: Mail Bonding with OSI Directory Services,* Prentice Hall, Inc., ISBN 0-13-683210-5, 1992.

STEVE HARDCASTLE-KILLE is President of the ISODE Consortium, which is a not-for-profit cooperative enterprise, whose mission is to promote and develop the ISODE package, including the PP X.400 System. He has a degree in Physics from Oxford University, and degrees in Electrical Engineering from UMIST (Manchester) and Stanford. Until recently, he was a Senior Research Fellow at University College London, where he has worked for the past ten years on various aspects of networking and distributed systems. He is the architect of the PP and QUIPU systems, and chairs the IETF WG on OSI Directory Services.

The PP Message Transfer Agent

by S. E. Hardcastle-Kille

Introduction

This article describes PP, which is a state of the art MTA, with a range of capabilities. A *Message Transfer Agent* (MTA) is a message switch which provides store and forward capabilities for electronic mail. MTAs are the fundamental underlying components necessary to build a large scale distributed message handling service.

A key problem of message handling is that there are very many protocols which result in many single protocol islands with only patchy connectivity between them. In order to provide the highest possible connectivity it is important for PP to support multiple protocols and to provide this reason, PP is a multi-protocol MTA, and provides conversion between different protocols.

Message gatewaying is not an add-on to PP, but is a fundamental component. Two protocol families are of particular importance in PP: X.400 and RFC 822. X.400 is widely recognised as the standard which will be used as the backbone for the global message handling service, and PP is substantially oriented towards X.400. RFC 822 based systems are widely deployed with high interconnectivity, especially using the *Simple Mail Transfer Protocol* (SMTP) on the Internet. The management features in most existing MTAs are poor or non-existent. A major feature of PP is its management capabilities, which are critical for service deployment of message services.

History

Work on PP began in 1985 when a group of people wished to produce a system to support X.400. It was clear that existing RFC 822 MTAs were not going to be extensible to do all that we wanted, as RFC 822 assumes minimal features for message transfer, and X.400 has high functionality. We had looked at some existing X.400 systems, and these did not seem to be a suitable basis for our work. PP arose as the solution to the problems raised in these discussions. The ideas for the basic structure draw heavily on the MMDF system [8], and it is fair to say that MMDF is the parent of PP. The initial design and coding were a spare time activity, and early versions of PP were kept private to those working on it.

PP received its first public airing at CeBIT (Hannover Fair) in March 1989 under the auspices of the ESPRIT PODA project, where it carried ODIF over X.400, and converted it into local format by use of private conversion tools. It interworked with a number of X.400(84) systems at this point. The PP project gradually gained formal recognition, and received funding from a number of sources, the most significant being from the UK Joint Network Team.

PP is associated with the ISODE System, and makes extensive use of ISODE components [13]. In many ways PP is conceptually a part of the ISODE, but has been released separately to simplify checkout procedures. PP has been made openly available (effectively public domain), with the same conditions as for the ISODE. Public releases of PP were made available in September 1990 (PP 5.0) and more recently in December 1991 (PP 6.0).

X.400 support

The CCITT X.400 Recommendations on Message Handling Systems in 1984 and 1988 are the broadly accepted standards for building global electronic mail services [9, 10, 14]. PP supports most of the Message Transfer Service defined by X.400, and provides both the 1984 and 1988 versions of the Message Transfer Protocol (P1) to give remote access to this service.

The only major omission is of the public key (X.509) based security services. Key X.400 services supported include:

- Deferred Delivery
- Positive and Negative Delivery Reports
- Message Redirection
- Distribution Lists
- Probe
- Message Priority

Lower Layer flexibility

PP supports downgrading from 1988 to 1984 X.400, including a number of extensions to the standard, to allow any 1988 recipient to be addressed from a 1984 system. This downgrade specification is being progressed in the IETF and is a proposed standards track RFC [3].

PP provides a range of stacks to use X.400. There are a number of presentation layer and session layer options for X.400, and PP supports all of these. Support at the transport and network layers is dependent on the platform on which PP is running. Three common OSI Stacks are supported in many systems.

- Transport Protocol Class 4 on Connectionless Network Service
- Transport Protocol Class 0 on X.25 or Connection Oriented Network Service
- Operation using a mapping of Connection Oriented Transport Service onto TCP/IP as specified in RFC 1006 mapping [12]

Operation of X.400 over TCP/IP is seen as a vital component of PP, and later versions are likely to also include a lighter weight mapping, assuming that an Internet Standard in this area is agreed.

SMTP and RFC 822

The RFC 822 protocols [2], and in particular SMTP (*Simple Mail Transfer Protocol*/RFC 821) are a key component of PP [11]. These protocols are very widely deployed in the research community, where PP has its origins. PP has a full implementation of these protocols, which are fully conformant to the Host Requirements and the other RFCs implied by this [1].

Mapping between RFC 822 and X.400 is a key issue addressed by PP. Specifications of this mapping have been made in parallel with its implementation in PP, and this work is expected to be a proposed Internet Standard [5].

Protocol extensibility

PP is designed so that it is straightforward to add in additional mail protocols, by writing protocol specific "channels." Because of the protocol independent manner in which PP is written, adding in a new channel provides a gateway from that protocol to all of the other protocols supported by PP. Channels which have been added to PP include:

- UUCP Mail
- JNT Mail (the UK Academic Community Mail Protocol)
- DECnet Mail-11
- At least three other vendor specific protocols

continued on next page

The PP Message Transfer Agent *(continued)*

Access units

X.400 defines the concept of an *access unit,* which provides a mapping onto services which are not directly equivalent to the Message Handling Service. The standard bodies have defined access units for: Teletex; Physical Delivery (i.e., Paper Mail); and Telex. CCITT/ISO are working on a mapping for Facsimile, but prior to this being available, PP provides a (non-standard) access unit onto Facsimile (G3 Fax), and support for two makes of fax modem is offered. It is expected that support for more modems will be offered in future releases. This access unit provides a useful service, which can be easily accessed from either RFC 822 or X.400. It also provides a model on which other access units could be based.

Content conversion

In addition to the protocol conversion already mentioned, PP provides facilities for conversion of the content of messages. For example, it can map between text body parts in different character sets. This is an important service, as it allows for new formats to be introduced into a message service, without requiring universal support. The lack of conversion facilities makes it very difficult to move beyond the use of the ASCII character set, except in closed environments. PP provides a general purpose mechanism, whereby content types (X.400 encoded information types) can be configured in easily, and mapped by filters which do not need knowledge of message formats.

Configuration

PP can be extensively configured, to allow it to be used for a wide range of purposes. The configuration of a PP system is controlled by a tailor file, which sets parameters, and defines the selected components (e.g., which protocols are supported). Many functions of PP have associated tables, which are referenced from the tailor file. Use is also made of OSI directory services, and name services such as the domain name system.

Authorisation

PP provides mechanisms to control which messages are authorised to be sent, for example, to restrict some users to local access only, or to allow selected users access to services which incur significant costs. A basic unit of authorisation is the *channel,* which is a set of MTAs grouped together to enable clean definition of authorisation policy. Three types of control are provided:

- *Policy:* This allows a default policy to be specified, for example to allow free access to some channels and to prevent relaying between others.
- *Per user:* This allows controls to be set for a specific user to enable or disable sending and reception of mail by defined paths.
- *Per MTA:* This allows identified MTAs to be authorised, perhaps in conjunction with a bilateral agreement.

Information on each message and its authorisation is logged. By relating this information to the authorisation database, billing and accounting information may be generated, in addition to general usage statistics. To facilitate introduction and change of authorisation, a mechanism to send warning messages to users is also provided.

Management

The dynamic control of the MTA queue is provided by a single *queue manager* process (QMGR). This single process allows for sophisticated dynamic scheduling and control of messages. The scheduling control allows optimisation of MTA connections and control on other parameters such as message priority.

The QMGR provides a management port, which is access using a *Remote Operation Service* (ROS) based protocol. This port allows for monitoring and control of the MTA queue by use of an MTA Console, which is illustrated in Figure 1.

Performance

The PP MTA is designed to be able to support high volume message switching. Under low load, messages will be submitted, formatted and delivered in a few seconds, and the MTA will usually appear to be idle. The MTA queue can hold large numbers of messages. Queues of 10-20,000 messages have been reached, with this bound forced by available disk rather than a limit of PP. On a SUN Sparcserver 4/330 with SCSI disks, throughput of messages is up to about 30,000 messages per day (about 0.3 messages per second). Throughput is limited by synchronous disk I/O. A benchmark indicates that disk I/O alone would give an absolute limit of about 100,000 messages per day. Given the variations of real message traffic, the figures achieved relate to this. Hardware, such as the Prestoserve™, will improve the disk limit by a factor of about 10. It has not been possible to use this on an MTA with high load, to determine throughput for such a configuration.

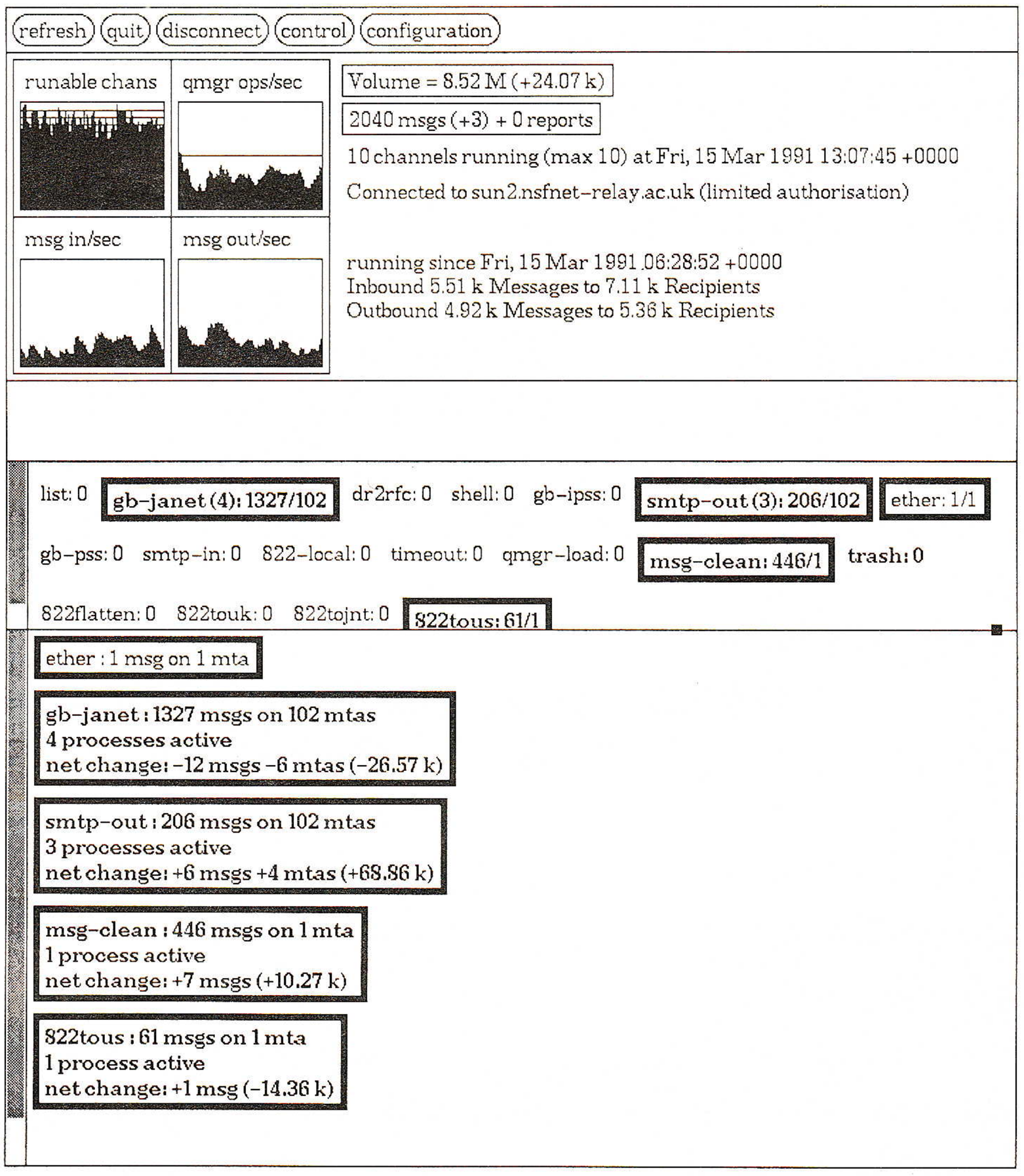

Figure 1: MTA Console

continued on next page

The PP Message Transfer Agent *(continued)*

Deployment of PP

Several hundred PP MTAs are in operation at sites around the world. PP is often used for gateway type services, and in particular for conversion between RFC 822 and X.400. It is also used at sites which switch large volumes of messages in a single protocol environment, that wish to benefit from PP's advanced management facilities. PP is used extensively in the RARE and COSINE X.400 operations in Europe, and in the Internet X.400 pilot. PP has also been used increasingly by organisations which use the MTA as a "mail hub," to give a uniform service across an organisation using a diverse collection of mail systems.

User Agents

PP is an MTA, and does not include User Agents. This was a conscious choice, and it is hoped that User Agents to go with PP will appear. Most existing RFC 822 based interfaces are supported, and many PP sites support users in this way. PP will also support the new MIME work [15] as an interface, with conversion to X.400 and content conversion available. Experimental X.400 User Agents have been written, and PP Application Program Interfaces are published in order to facilitate and encourage such User Agents to be written. There is ongoing work at X-Tel Services and University College London to produce a high end X Window System User Agent in conjunction with an X.400 Message Store.

Directory—The vital support

Use of the X.500 directory [16] is critical to effective deployment of X.400, and PP plans to make extensive use of the directory. Currently, the directory is used to support distribution lists. The most important use of directory will be to support routing and local configuration. The design effort for this is underway [4], and this document is expected to be used as a basis for work by the new IETF MHS-DS working group on message handling systems use of directory. An important related use is to determine user agent capabilities, and in particular the ability to support multi-media messaging. This function will be very important to enable smooth deployment of multimedia user agents.

Future directions

There are many things which are planned for the PP System. These include:

- *Use of Directory:* Described above.
- *SNMP:* Monitoring using SNMP, to provide uniform management of network components and application components. This would allow basic status checking of a large number of MTAs.
- *Privileged QMGR:* The current QMGR is not privileged, and so cannot (safely) initiate activities such as deleting messages. Such functions would be very useful, and it is planned to enhance the QMGR and console so that a range of operations to modify the queue are available.
- *Message Store:* It is planned to add a Message Store, which supports access by standard and extended P7 protocol [7].
- *Simplified Configuration:* The current MTA is complex to configure, and there will be a movement towards "plug and play."
- *Conformance:* The PP MTA has undergone extensive interoperability testing, but has not been conformance tested. Conformance testing will be undertaken.

Further information

The PP Manuals, together with a chapter on the architecture of PP and a design analysis is packaged in a book [6]. The manuals may be obtained separately with the PP software.

Getting the PP software

The PP system may be accessed by FTP from `cs.ucl.ac.uk`, `uu.psi.com`, or `archive.eu.net`. Retrieve the file `pp-6.tar.Z` in binary mode from the `isode/` directory. This file is the *tar* image after being run through the compress program and is approximately 3Mb in size.

References

[1] Braden, R., "Requirements for Internet hosts—Application and Support," RFC 1123, October 1989.

[2] Crocker, D. H., "Standard for the format of ARPA Internet Text Messages," RFC 822, August 1982.

[3] Hardcastle-Kille, S. E., "X.400 1988 to 1984 downgrading," August 1991, Internet Draft.

[4] Hardcastle-Kille, S. E., "PP use of directory," Research Note RN/92/14, Department of Computer Science, University College London, January 1992.

[5] Kille, S. E., "Mapping between X.400(1988)/ISO 10021 and RFC 822," RFC 1148, March 1990. Also available as MailGroup Note 23.

[6] Kille, S. E., *Implementing X.400 and X.500: The PP and QUIPU Systems,* Artech House, 1991, ISBN 0-89006-564-0.

[7] Kille S. E., "PPMS—The PP message store: Overall design," Research Note RN/91/5, Department of Computer Science, University College London, February 1991.

[8] Kingston, D. P., "MMDFII: A technical review," In *Proceedings of the USENIX Summer 1984 Technical Conference,* August 1984.

[9] Recommendations X.400, CCITT SG 5/VII, "Message Handling Systems: System Model—Service Elements," October 1984.

[10] CCITT Recommendations X.400 / ISO 10021, April 1988, CCITT SG 5/VII/ISO/IEC JTC1, "Message Handling: System and Service Overview."

[11] Postel, J. B., "Simple Mail Transfer Protocol," RFC 821, August 1982.

[12] Rose, Marshall T., & Cass, Dwight E., "ISO Transport Services on top of the TCP," RFC 1006, May 1987.

[13] Rose, Marshall T., "The ISO development environment: User's Manual (Version 6.0)," January 1990.

[14] Onions, J., "Components of OSI: X.400 Message Handling System," *ConneXions,* Volume 3, No. 5, May 1989.

[15] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1341, June 1992.

[16] Benford, S,. "Components of OSI: X.500 Directory Services," *ConneXions,* Volume 3, No. 6, June 1989.

STEVE HARDCASTLE-KILLE is the president of the ISODE Consortium. Prior to this, he was a Senior Research Fellow in the department of Computer Science at University College London. He has a BA and MA in Physics from Brasenose College, Oxford, an MScs in Electrical Engineering from University of Manchester Institute of Science and Technology and Stanford University. He has worked at University College for the past ten years, and has conducted research in a range of areas relating to networking, messaging, directories, and distributed systems. He is the architect and project manager for the *PP* (X.400) and *QUIPU* (X.500) systems.

Electronic Data Interchange (EDI) over X.400

by Sara Radicati, Creative Network Solutions

Overview

Electronic Data Interchange (EDI) is one of the fastest growing applications of X.400 Message Handling technology. Several approaches are available today for deploying EDI over X.400 message handling networks. EDI transactions can be transferred over 1984 or 1988 X.400 implementations by simply encapsulating the EDI transaction into a P1 envelope, or within the body of a P2 interpersonal message. Alternatively, a new protocol, called *P-edi,* has been defined by CCITT specifically for use in transferring EDI information. P-edi is intended to work primarily with 1988 X.400 implementations, and is a complete replacement to the P2 protocol. This article describes some of the techniques which are available today for implementing EDI over X.400 message handling systems and discusses what are the key issues in planning and deploying X.400-based EDI systems.

EDI provides an automated way for businesses to exchange structured business data such as invoices, purchase orders and other financial documents. While EDI has been around for a while it has traditionally relied on Bisync types of communications or even simply the recording of business data on magnetic tapes for later transfer by non-automated means.

Although EDI and X.400 have developed independently, the asynchronous nature of EDI transactions makes them ideally suited for transport over X.400-based message handling systems. EDI today is one of the most interesting and commercially viable applications of electronic messaging.

Two approaches

Two interim approaches have been developed to date for transporting EDI transactions over X.400 message handling networks. The first, called the NIST P0 approach, is more commonly used in North America. The second approach, called P2, is more commonly deployed in Europe and is consistent with the EDIFACT standards and guidelines.

Both the P0 and the P2 approach are being superseded by a new ISO/CCITT protocol standard, called P-edi (recommendation X.435). P-edi was developed specifically to support the deployment of EDI over X.400 networks. It offers two major advantages over traditional EDI offerings and the P0 or P2 based EDI approaches: support for 1988 X.400 Security and Authentication, and enhanced receipt confirmation and notification services.

Value Added Network providers (VANs) today offer primarily P0 or a P2 based EDI services. The P0 and P2 approaches will most likely continue to be around for at least the next 3–5 years. However, VAN providers are expected to begin to roll out P-edi based services alongside their current P0 or P2 services within the next 12 to 18 months.

Companies wanting to do EDI over X.400 today, have a choice of adopting a P0 or a P2 approach now, or waiting until P-edi services are more readily available and migrate their current business processes directly to P-edi.

Migrating network installations forward from a P0/P2 based approach to P-edi is expected to be non-trivial, as it will require the deployment of completely new front-end systems to deal specifically with P-edi formats and protocol functionality.

In particular, P-edi front-ends will need to be designed to deal with the fact that a P-edi message can contain multiple body parts including graphics and text which need to be routed and processed differently from the primary body part containing the EDI transaction.

EDI considerably expands the market for X.400 communication products and therefore represents a major opportunity for vendors of X.400 products. However, X.400 vendors need to offer front-ends which can intelligently route and process EDI transactions.

Users looking to deploy EDI services should seriously consider using EDI in conjunction with X.400 services because of the global communication infrastructure which X.400 provides. While P0 and P2 based approaches will be around for a long time to come, users in industries that are just beginning to deploy EDI services should try to take advantage of the new P-edi features and services as they become available. P-edi's enhanced support for security and end-to-end receipt notification makes it the best long-term solution for dealing with the delicate nature of EDI financial transactions.

Benefits of using EDI

The central idea behind EDI is that it should be possible to automate the way in which companies do business with each other through the exchange of purchase orders, invoices and other structured business data. Given that the information conveyed by such business forms typically originates on a computer and is usually processed by a computer at the receiving end, it is highly preferable to transfer the information directly in electronic form between computers rather than having to rely on the exchange of paper forms. Using computers and computer-to-computer communications to handle business transactions can cut down considerably on the time required to process business data. For example, a purchase order transmitted to several suppliers can be transmitted and processed more quickly if it is sent electronically. This in turn can result in faster turn around times in filling inventory and meeting delivery schedules.

In essence, EDI can substantially streamline business interchanges by reducing the number of paper forms and documents which must be exchanged between two organizations to complete a business transaction. This in turn leads to higher productivity, better inventory and resource management, as well as substantial cost savings.

However, in order for trading partners to successfully exchange business transactions electronically, all of the companies involved in the exchange must share a standard format to represent their particular business transactions. EDI standards have emerged in recent years both in the US and in Europe, which define both the syntax and the semantics of these transactions.

EDIFACT

In the US, the most commonly employed EDI format is ANSI X12. In Europe the EDI standard is based on ISO standard 9735—and is called EDIFACT (*Electronic Data Interchange for Administration, Commerce, and Transport*). Another common EDI standard is that defined by the United Nations, called UNTDI (*United Nations Data Trade Interchange*). In addition, more specific conventions and guidelines for EDI have been defined within specific industries such as the aircraft manufacturing industry and the automobile industry.

To date EDI standardization work has focused primarily on the information formatting aspects of the business exchange, rather than worrying about the underlying communication requirements.

continued on next page

EDI over X.400 *(continued)*

All the above standards deal exclusively with the way in which business information is structured and make no assumptions regarding how it is transferred between computers. Traditional communication of EDI transactions has typically taken place through ordinary data transfer mechanisms, ranging from mailing computer tapes to the use of private leased lines between host computers.

The X.400 store and forward technology provides a unique platform for the automated deployment of EDI. First of all X.400's store and forward nature is inherently well suited for EDI kinds of transactions which do not require real time processing. EDI messages are primarily generated by computer processes, although some human intervention may also take place. Second, the fact that an international messaging infrastructure based on X.400 technology is rapidly becoming available in most industrialized nations around the world. This makes X.400 the ideal information transfer mechanism to use for doing business in a global economy.

Interim approaches to EDI over X.400

The 1984 and 1988 X.400 message handling standards were initially developed without any specific thought for the support of EDI. Since then, special provisions have been established by NIST and other functional standards groups, for the interim deployment of X.400 protocols to support EDI exchanges.

Unfortunately the US and Europe have tended to adopt different approaches to EDI over X.400. In the US, the work on EDI was led primarily by NIST and reflects very closely the way in which EDI was being used in North America at the time. Transactions consist mostly of large exchanges of information containing many separate transactions, usually occurring at low intervals between trading partners. In Europe, the trend has been to use EDI in a more distributed fashion to handle short, frequent transactions between computers.

The NIST approach

NIST's approach to the problem was to meet primarily the needs of the North American market. Consequently the ANSI X12 version of EDI was adopted. NIST felt that the P2 interpersonal messaging protocol was not well suited for EDI since it is designed to handle mostly small messages, exchanged between human users which may not require extensive confidentiality services. EDI as it is used in North America, on the other hand, requires the exchange of large documents between companies or central EDI communication agencies (often referred to as "Clearinghouses"), as well as a high degree of confidentiality.

NIST, therefore, chose to handle EDI transactions in the X.400 architecture at the message transport (MTS) level. The NIST approach supports the embedding of EDI messages within the content field of the P1-level X.400 message. It uses the undefined content type 0 to indicate to the receiving end that the content of the P1 message is an EDI interchange. The NIST approach has been dubbed the P0 approach. The P1 messages are then transferred transparently between MTAs as is the case for all X.400 application protocols. The P1 message is re-constructed at the receiving end only if it comprehends messages of content type 0. Otherwise, the message is not delivered.

European organizations approached the problem differently, largely because EDI is used differently in Europe than in the US. EDI usage in Europe makes less use of large single transfer agents but relies more on smaller direct EDI exchanges between computers.

European designers decided to embed EDI messages within P2 interpersonal messages and make use of some of the P2 fields to carry EDI header information. The P2 approach has the advantage that EDI interchanges can also be annotated by text, graphics and any other valid P2 body types. Figure 1 below shows the differences between the NIST P0 approach and the European P2-based approach.

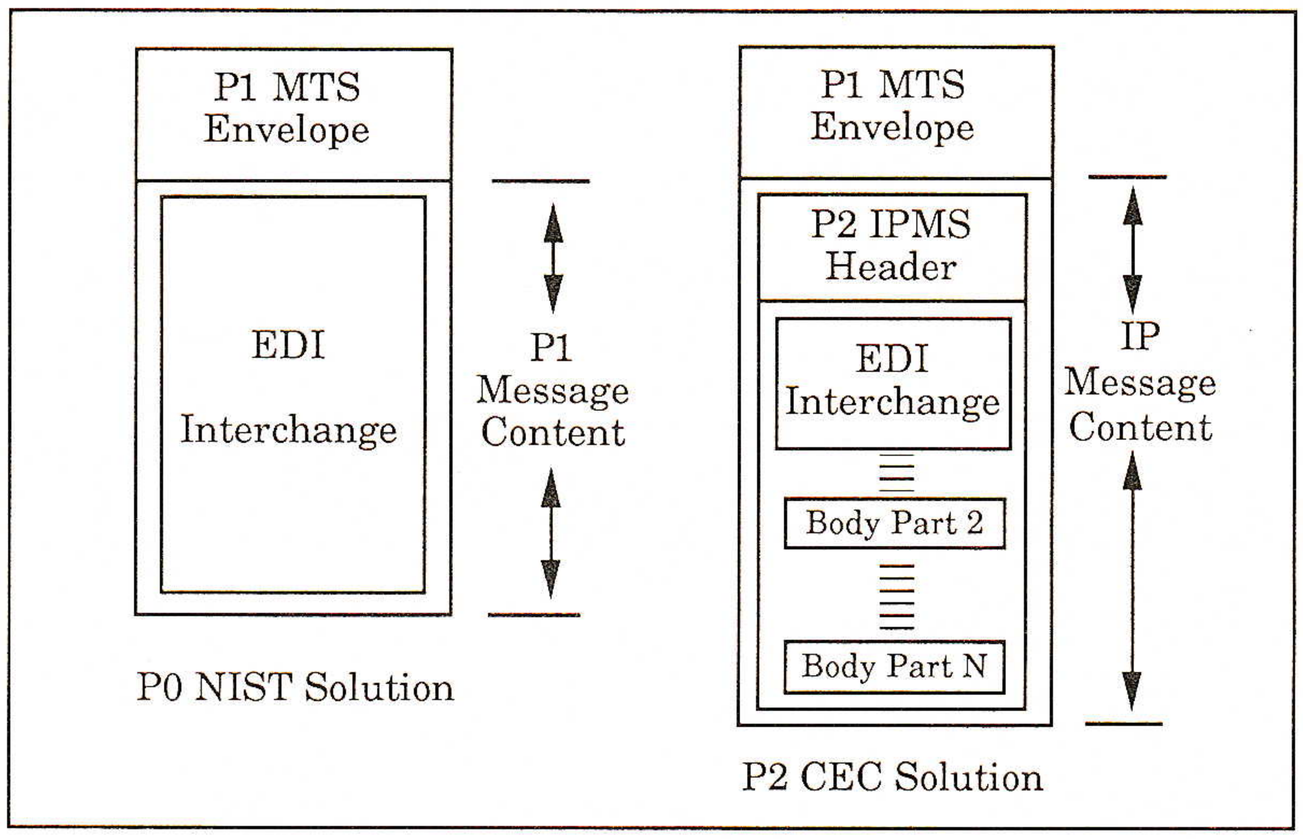

Figure 1: NIST P0 versus the European P2 approach

TEDIS guidelines

In order to harmonize these differences and better promote the use of EDI, the European Community under the TEDIS (*Trade Electronic Data Interchange Systems*) program has established a number of guidelines for the use of EDI in conjunction with X.400. These are known as the "TEDIS Guidelines," and have been defined in cooperation between European and North American users and experts.

The TEDIS Guidelines confirm the use of P2 interpersonal messages and recommend the continued use of this approach in Europe. However, they also strongly discourage the use of P2 protocol service elements in implementing the P2 approach so as to make it easier to construct gateways which will map EDI transactions from the P0 into the P2 format and vice versa. This makes it easier for EDI communications between Europe and the US to occur through the use of simple gateways. The overall intent in the TEDIS Guidelines was that both the NIST and the European approach will eventually be superseded by an international ISO/CCITT standard defined specifically to support EDI over X.400.

The X.435 P-edi standards

EDI was one of the major new work items under development during a CCITT 1990 accelerated study period. The 1990 message handling work has developed a new X.400 application protocol called P-edi to specifically support EDI data interchanges. P-edi represents a new message content type that has been specifically optimized to meet the needs of EDI transactions. The CCITT work on P-edi has been ratified as recommendation X.435, "MHS: EDI Messaging Systems." An F-series document has also been published by CCITT, called F.435, "MHS: EDI Messaging Service."

1990 MHS defines an architectural model which embeds EDI applications within the MHS environment. Figure 2, on the following page, illustrates how the P-edi protocol fits into an X.400 architecture relative to the other message handling protocols.

continued on next page

EDI over X.400 *(continued)*

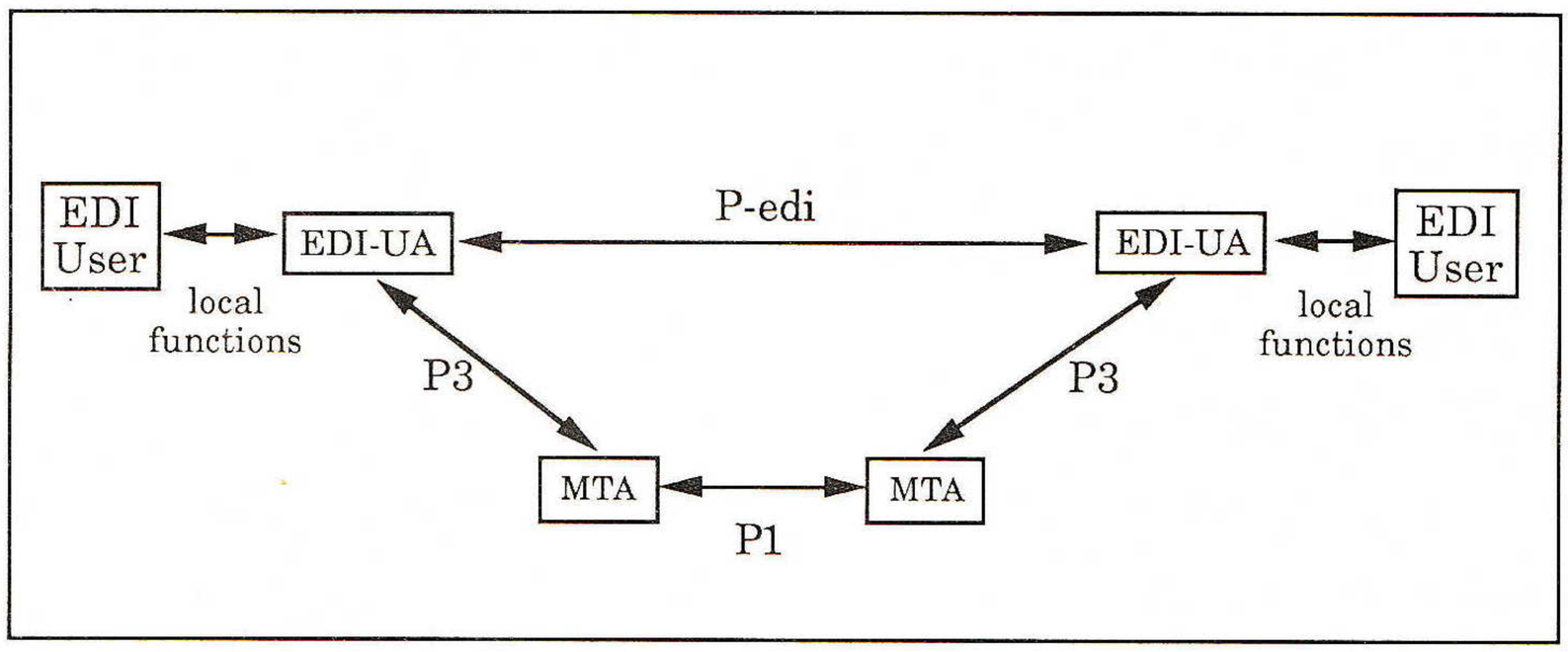

Figure 2: The P-edi Protocol in MHS

The EDI Model

X.435 defines the concept of EDI user agents (EDI-UAs) which are analogous to the P2 interpersonal user agents (IPM-UAs) defined by 1984 and 1988 X.400. EDI-UAs create messages (EDIMs) with a content type specific to EDI. The heading of an EDIM contains information fields which are present in the EDIFACT interchange header segments (or corresponding ISA or STX segments for ANSI X12 and UNTDI EDI exchanges). It also carries additional service requests which may have been set by the message originator. 1990 MHS also defines special *Message Store* facilities specifically tailored to the needs of the EDI-UA users. One of the drawbacks of P-edi is that it assumes a 1988 X.400 message handling architecture in order to provide security and end-to-end receipt notification. Unfortunately, 1988 X.400 products and services are only slowly beginning to emerge on the market. Figure 3 shows the architecture of an MHS environment which includes EDI services.

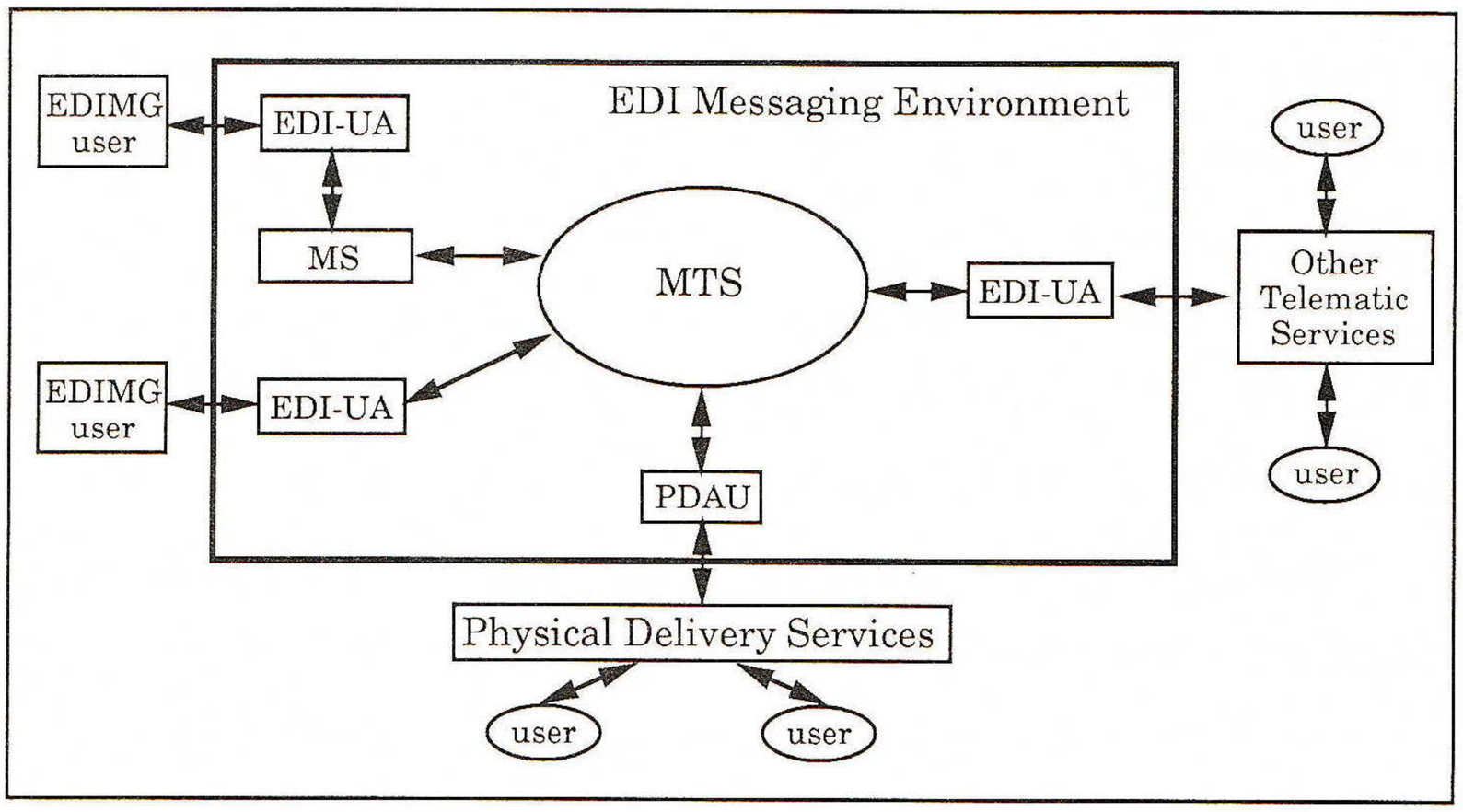

Figure 3: The EDI Messaging Environment

EDI names

EDI messages are routed within the MTS on the basis of their O/R Address in the same way as any other X.400 message. In addition, EDI messages may carry naming information which is specific to the particular EDI environment. EDI communities can be organized by:

- Industry groups,
- Private trading groups of a large corporation, or
- Third-party EDI service providers.

The EDI names used by any of these communities may be either issued by an international naming authority, by a multinational company, or free-form names assigned by the trading partners themselves. EDI names contain no geographic element. The EDI standards allow the use of a qualifier with the EDI name in order to denote the naming authority which assigned the alphanumeric string that makes up the name. The use of the qualifier in conjunction with the EDI name makes it possible to achieve globally unique EDI naming.

Information about EDI users can be stored in a Directory along with other useful information about network users (e.g., X.400 users and resources). A new object class called *EDI-user* is defined to contain attributes for the name of the EDI user and the capabilities which pertain to particular users. The capabilities which may be associated with an EDI-user entry include information about what EDI standard is supported, what EDI character set, as well as any specific versions or releases of EDI documents that are supported.

EDI messages

EDI messaging (EDIMG) allows the exchange of 2 kinds of messages:

- *EDIMs:* EDI messages which contain the actual business transaction(s), and
- *EDINs:* EDI notifications containing information pertaining to the successful or unsuccessful delivery of an EDI interchange. These are analogous to receipt notifications in P2.

The Heading component of an EDI message corresponds to a sequence of information fields which characterize the EDI message, whereas the Body component is a sequence of one or more Body parts. The body of an EDIM is defined to consist of what is called a *Primary Body Part* which contains the EDI information object. The primary body part may be either an EDI interchange or a forwarded EDIM. EDI interchanges can contain EDI information objects which are defined according to the EDIFACT, the *United Nations Data Trade Interchange* (UNTDI), or the ANSI X12 standards.

Additional body parts may be present beyond the primary body part in an EDI message, but these can contain only different (non-EDI) body types which are associated with the EDI interchange. This information may include textual information, voice annotations, graphics, etc. Externally defined body parts can also carry information objects whose semantics and abstract syntax (i.e., their ASN.1 definition) are identified by an object ID which is carried with the body part. The standard explicitly states that externally defined body parts may not include other EDI interchanges, since only one EDI interchange is permitted per message. Figure 4 depicts the structure of an EDI Message. EDI can be used to send information to physical delivery units for postal delivery, as well as to facsimile machines.

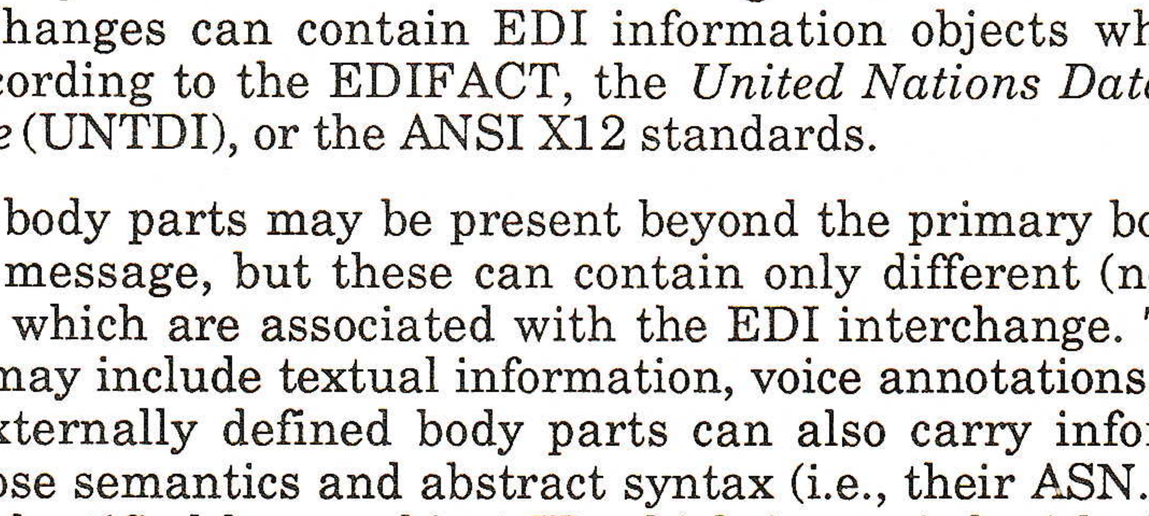

Figure 4: EDI Message Structure

continued on next page

EDI over X.400 *(continued)*

EDI notifications

The 1990 MHS concept of "EDIM responsibility" provides a method for tracing the "acceptance" of EDIMs as they leave the MHS environment. Responsibility for EDIMs may be either: *accepted, refused* or *forwarded.* The element of service "EDI Notification Request" allows an originator to request notification from each recipient EDI-UA indicating whether the EDIM was accepted, refused or forwarded. Notifications can be requested independently, or on a per recipient basis, or the request can be omitted altogether.

It is often desirable for organizations to receive all EDI messages at a central location, or EDI User Agent, which will then forward them to the final destination EDI-UAs. This is particularly useful in large organizations where it enables functions such as logging and auditing of all EDI message traffic at a central location. An EDIM may therefore be accepted by an intermediate EDI-UA before being forwarded on to the final intended recipient EDI-UA.

P-edi advantages

P-edi offers two major advantages over traditional EDI offerings, as well as the P0 or P2 based EDI approaches, these are:

- Security, and
- Enhanced receipt confirmation and notification services

P-edi security is largely based on the security and authentication mechanisms already defined for 1988 X.400. These include digital signatures, the ability to authenticate senders and receivers, message encryption, as well as message origin authentication checks. In addition, the new notification mechanisms described above provide a higher measure of delivery assurance and end-to-end acknowledgements not available to date with any other EDI transport service.

Interworking issues

Interworking between the NIST P0 approach and the EDIFACT P2 approach is largely taken care of through the TEDIS Guidelines mentioned earlier. By ensuring that P2-based EDI messages utilize only the body of the message and do not appropriate standard fields in the P2 header, it becomes fairly straightforward to develop gateways which can repackage EDI transactions from a P2 format into a P0 format, and vice versa.

Most VANs today offer either a P0 or a P2 based EDI over X.400 service. The P0 and P2 approaches are likely to be around for a long time to come, at least on the order of the next 3–5 years. VAN providers are expected to begin to roll out P-edi based services alongside their current P0 or P2 services within the next 12 to 18 months.

In terms of EDI application products (i.e., EDI user agents), the market is split about evenly between front-ends which emit P0 and front-ends which emit the P2 format. Vendors like HP and DEC already offer EDI user agents in conjunction with their X.400 products. Frequently, however, customers still need to do a sizable amount of development on their own to interface their in-house business management systems to an X.400 EDI front-end.

Interworking between the new P-edi protocol and the current P0/P2 EDI approaches is likely to be much more complex than simply translating between P0 and P2. Since P-edi is a completely new application layer protocol it does not map well to either the P2 or the P0 approach. In particular, P-edi introduces concepts of security and end-to-end receipt confirmation which are simply not available in the P0 and P2 implementations.

Although gateways will be available to convert between P-edi and P0/P2, the P-edi security and notification enhancements will be lost when converting from P-edi to a P0 or a P2 format.

Migrating a network installation forward from a P0 or P2 based approach to P-edi will be non-trivial as users will need to deploy completely new front end systems to deal specifically with the P-edi format and protocol functionality. It is expected that for the most part P-edi services will grow alongside the currently provided P0 and P2 based services, and that users will choose to transition their implementations over gradually to the new P-edi services.

Front end systems need to become much more sophisticated to handle P-edi. In particular, they will need to be designed to deal with the fact that a P-edi message can contain multiple body parts including graphics and text which need to be routed and processed differently from the primary body part containing the EDI transaction.

Strategies for EDI deployment

EDI represents a major change in the way in which companies do business. Users who are thinking of deploying EDI services within their companies should expect to convert their paper-based business processes over to EDI gradually. Most likely they will find that they need to initially build an EDI infrastructure alongside the traditional paper-based business processing structure, and then gradually transition suppliers and applications over to EDI.

A strategy for deployment of EDI services should carefully take into account the extent to which business partners' and suppliers are also migrating to EDI and what types of EDI services they are selecting. Large companies will have an advantage in terms of being able to set the course for most of their smaller suppliers and strongly influence the choices of their principal trading partners. To a large extent users' choices will be guided by what services are offered by VANs, as well as by the kinds of services their suppliers and trading partners are able to deploy.

Companies considering EDI over X.400 today have several choices: they can adopt a P0 or a P2 approach right now, or wait until P-edi services are more readily available and migrate directly to P-edi.

To the extent that an industry or a segment of an industry is embarking on EDI for the first time, the preferable choice is to go directly to a P-edi implementation. The P-edi approach will eventually be best supported by VANs because it is an internationally endorsed standard. Also from a technical standpoint, P-edi offers a much higher level of service than is currently available through either bisynchronous communications or the use of P0/P2 X.400-based approaches. The addition of enhanced security services and greater delivery acknowledgement and confirmation services makes P-edi much better suited to the delicate financial nature of EDI transactions.

On the other hand, P-edi services may be slow to materialize. While we expect that VANs will make such services available in the near future, the issue of P-edi capable front-ends remains. In order to take full advantage of P-edi's value-added features, a new generation of more capable front-end P-edi products needs to become available on the market. Alternatively, users will be left to deal with the complexity of designing and implementing such front ends on their own.

[Ed.: This article is reprinted with permission from *The Messaging Technology Report,* April 1992.]

SARA RADICATI is president of Creative Network Solutions. She has been involved with the development of electronic mail technologies and standards for the past ten years. She has been a principal contributor to the design of the X.500 standards for Directory Services and has participated in major standardization forums, including ISO, CCITT, ANSI and NIST. Dr. Radicati was most recently responsible for Messaging Products and Strategies at Novell, Inc. Prior to joining Novell, Dr. Radicati was responsible for product architecture at Xerox Corp. She holds M. S. and Doctor of Science degrees from the California Institute of Technology and the University of Pisa, Italy.

Electronic Mail in Zambia

by Mark Bennett, University of Zambia

Background

The use of electronic mail in Zambia appears to be a recent phenomenon compared to some other countries in the region (for example Zimbabwe or Kenya), but the telecommunications infrastructure is sufficiently stable and advanced for the steps which are now being taken to give rise to optimism for its quick growth in the future.

As in most countries, there have for some time been individuals, particularly expatriates, who have been using their own computers to dial in to e-mail services elsewhere in the world, especially through the UK. A number of foreign journalists also use such services, as was witnessed during last year's elections in the country. Equally, organisations with international connections are starting to make some one-to-one connections (e.g., some development agencies—such as those now involved in assisting with the drought crisis—and research bodies), but most are still looking to a centralised service to be provided.

This short report will describe the two e-mail/conferencing systems being developed separately by the University of Zambia and by a group of non-governmental organisations. Both systems are *FidoNet*-based and are using *FrontDoor* software. In addition, reference will be made to the *HealthNet* satellite-based communications system being introduced for medical personnel.

Telecommunications infrastructure

The telephone system within Lusaka and the cities of the Copperbelt are relatively good by regional standards, with a growing number of digital exchanges now in operation. Phone connections within Lusaka are acceptable for data communications, and although the rainy season brings its problems these are generally of a short-term nature.

International connections vary depending on the destination. Lines to the UK and USA (being carried on a direct satellite link) are of a good quality, and although traffic is congested from certain exchanges, there is no problem in obtaining data quality lines (of at least 2400 baud) to the UK outside working hours. Lines to Harare, Zimbabwe, are also acceptable for data traffic and relatively easily obtained. Lines to South Africa are more difficult to obtain, but South Africa can make contact with Zambia much more easily (incoming international calls experience less congestion due to switch allocation). Again data rates of 2400 baud are regularly used and higher rates up to 9600 baud are expected to be possible.

Lines to Kenya (which are routed through Tanzania) suffer from considerable congestion. They can be obtained during the day through the operator, and occasionally at night by IDD for computer use. Quality is variable. Determination of other possible routes has yet to take place, but the above mentioned form the most useful links and could in turn be used to route messages elsewhere. It is not expected that other regional links will provide easy connections: some countries do not have international IDD; some have links with Europe based on infrastructures put in place during the colonial period. Hence lengthy routing.

Cost

In any instance, the major drawback on the use of the international telephone network from within Zambia is the cost, which has been artificially inflated by the PTC to discourage outgoing calls and the consequent loss of foreign exchange. A one minute call to Europe is charged at the local equivalent of US $7 per minute (there being no cheap rate period). Regional calls are about half that figure.

Since this means that a senior computer programmer at the University, for example, could be employed for one month for the same cost as a 15 minute phone call to the UK, there is clearly a considerable disincentive to unbridled expansion of conventional e-mail systems, and maximising the number of incoming calls (rather than outgoing) is a priority. Experimentation with 9600 baud lines to South Africa is also due to begin shortly as another potential cost-saving measure.

There is no packet-switching in the country at present, although the Zambian PTC have recently undertaken a study (in conjunction with British Telecom) to determine the demand for such facilities, and are, in outline, committed to the eventual introduction of X.25. This should also lead to a downturn in costs to e-mail users, although the priority for the PTT is clearly to provide service for the commercial sector.

ESANET

The University of Zambia (Unza) is the major university in the country, with 9 main faculties; a full-time student population of some 4,500 and an academic staff of around 450.

The University has joined an IDRC (Canada) funded research project to establish the feasibilities of electronic communication with a five university grouping within the region; the other universities being those of Zimbabwe, Dar-es-Salaam (Tanzania), Nairobi (Kenya) and Makerere (Uganda). Several workshops have been held to map out the strategy for the system to be known as *ESANET* (Eastern and Southern African Network). The system is to be based around one PC (as the "hub") at each university.

The optimum channels of communication have yet to be established, and not all sites are operational, although they should be by mid 1992. However, Unza has been running a trial system for some 9 months, and has launched it "officially" for use by the academic community. This has also included the holding of a number of seminars and publishing newsletters and information on the advantages and uses of the system.

GreenNet

Originally all mail was routed through *GreenNet* in London and the gateways there allowed the University to have an Internet as well as a *fido* address (mail coming though the *gnFido* gateway). This proved effective but was very expensive to run due to the high telephone costs.

The *GreenNet* connection has now been phased out in favour of routing through Rhodes University in South Africa. The political climate had originally ruled out *ESANET* links through South Africa but the gradual dismantling of apartheid has led the Commonwealth leaders (and later the UN) to accept the desirability of academic links, and the *UNINET* system in South Africa formally approached Unza with a view to cooperation in promoting regional information exchange. (*UNINET* is an Academic and Research Network.)

Rhodes University, on behalf of *UNINET*, then generously offered (at least on a medium-term basis) to bear the cost of sufficient polls each day to allow picking and dropping of mail which would be routed either into *UNINET* or to the wider Internet (or, of course, to *FidoNet* and the APC networks). Rhodes are currently polling some three times a day and effectively providing a free e-mail service to Unza.

The University of Zambia operates *FrontDoor* software and its Internet address is `jdoe@f1.n761.z5.fidonet.org`.

continued on next page

Electronic Mail in Zambia *(continued)*

With the service operating at 2400 baud a throughput rate of some 220cps is averaged (data compression also being used to improve the base figure). It is hoped that the introduction of a 9600 Trailblazer in the very near future will bring costs down substantially.

At present, all messages are centrally handled through one dedicated PC (with its own telephone line available 24 hours a day) housed in the University Computer Centre. Trials began with two University Schools; the School of Engineering, located on main campus; and the School of Medicine located at the University Teaching Hospital some 10km distant. Since there is as yet (for reasons of cost) no campus network in place normal phone lines were used (the School of Engineering having manufactured a device that allows the central PC to be connected to both internal and external phone systems and answer whichever calls first) and the systems run *FrontDoor* as *Fido* "points."

These trials proved most successful with staff quickly realising the potential that e-mail offered for international communication, and messages to and from five continents were soon being exchanged. Sufficient modems have now been obtained for all Schools, research and administrative units to be connected up (bringing the total to 20+). All but a few have now been put in place, with several Units simply having to wait until they can get access to a PC.

The PCs being used clearly do not need to be dedicated to this use, but only be available for a few minutes each day for picking and dropping of messages to the Computer Centre. The cost of day-to-day operation is therefore minimal.

Economics

Whilst this prototype system works well and messages have been sent and received worldwide to the benefit of a select set of staff it will be a number of months before the impact on the University as a whole is felt. The expectations of the academic community are considerable and the needs are great. The general economic recession, in particular the 10,000% rate of devaluation that has occurred over the last 6 years, has ensured that foreign travel has greatly reduced. This in turn means that the number of postgraduate students (needed as potential future staff members) has lessened dramatically. Collaborative research has also declined, and yet many of the difficulties caused by lack of mobility could be redressed if communications with colleagues/supervisors overseas could be improved and if these communications systems could also be used to provide access to up-to-date literature, itself a further victim of economic decline.

For e-mail systems to succeed, they must remain free of charge to academic staff. Any costs would be a stumbling block to their potential use. The *Foundation for Research and Development* (FRD) who run *UNINET* in South Africa have asked Unza to look into the possibility of a leased line service (running UNIX and TCP/IP). The cost from the Zambia end would be US $17,000 per year, plus a monthly subscription (R3,000) to *UNINET*. These costs would likely prove too high. It might only be hoped that if the project can be proved to be viable (and it is undoubtedly necessary) then outside funding may be attracted.

Until the pilot *Fido* system has been in place for at least 6 months it is very difficult to estimate the volume of traffic that may pass through the system, as the knowledge of the potential of e-mail is yet to be established in the minds of many of the staff.

Equally, the *Fido* system has its limitations against a leased-line alternative. The economic climate has meant a considerable shortage of books and journals coming into the Libraries. The possibility of literature searching and joining in USENET conferences through the Internet must prove most attractive, but may prove too much strain on a dial-up Fido line.

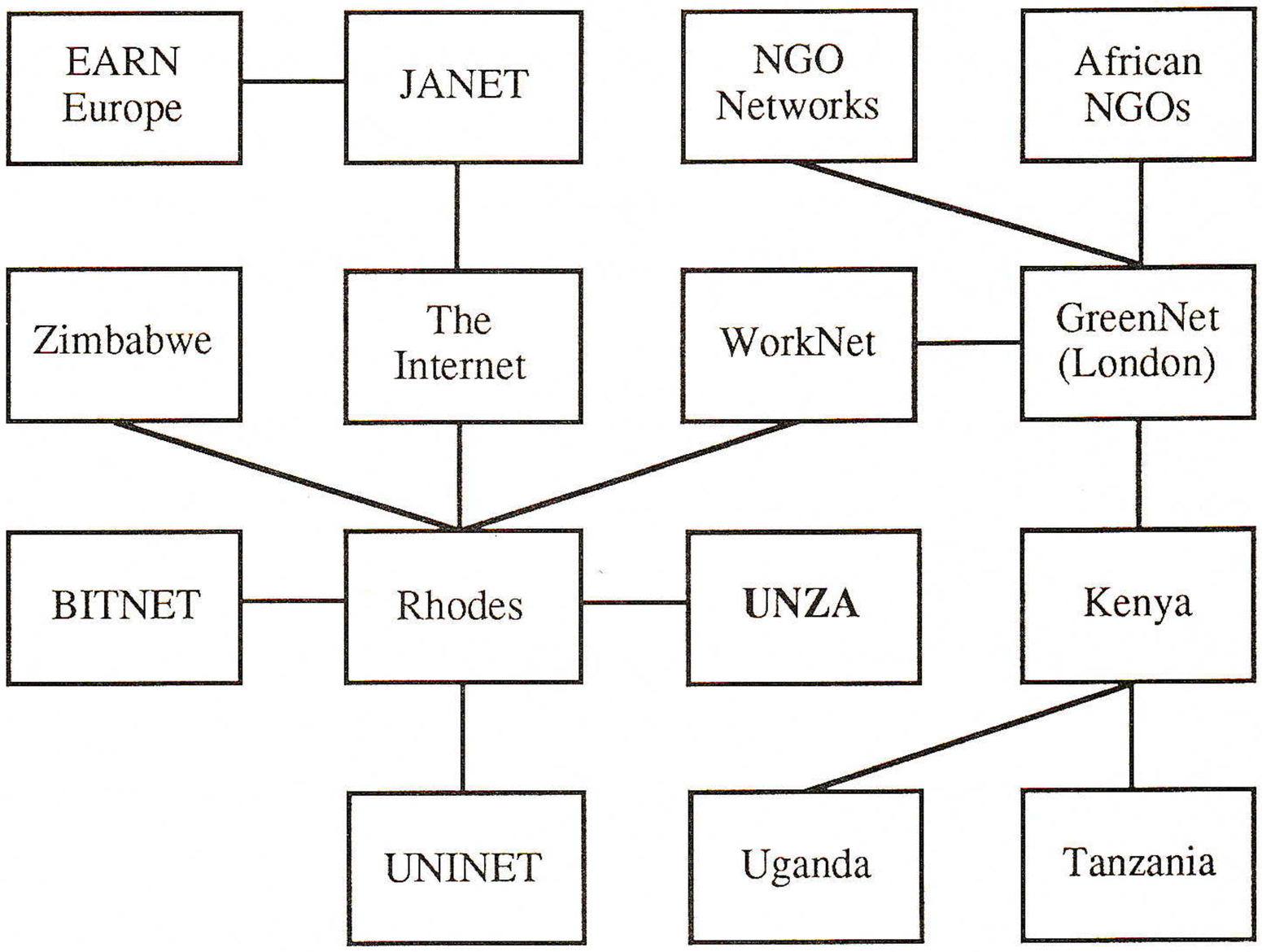

Figure 1: UNZANet international connections

Regional communication must also prove the subject of further research within *ESANET*. Figure 1 represents the schematic University links as seen from Zambia. It is noted that Zimbabwe is part of the *ESANET* project, and is also pursuing independent links with *UNINET* using UNIX. They have an official Internet address, even if dial-up lines are still used.

The University of Zambia, however, is greatly indebted to Rhodes University and *UNINET* for this opportunity to become one of the first sub-Saharan African universities (outside South Africa itself) to be able to offer e-mail access to the worldwide Internet and looks forward to all the possibilities which that might offer.

HealthNet

The University of Zambia is also taking part in another related scheme (along with the other *ESANET* participants) to provide a satellite based e-mail and literature searching system for health-based workers. Funded by *SatelLife* (a joint USA–Russian venture based out of Cambridge, Mass.) this system uses a low-orbit (850km) sun-synchronous satellite which passes over each location three or four times daily as a means of storing and forwarding messages. Because the satellite is in relatively low-orbit only cheap home produced antennae are required and a standard amateur-radio transceiver (plus a radio modem) provides the interface to the computer. The cost of providing a complete earth station is around US $5,000. Since this equipment is provided by *SatelLife*, the cost to the health staff for using the system is zero and the reliability should be high (compared to terrestrial based communication).

continued on next page

Electronic Mail in Zambia *(continued)*

This equipment has again been installed at the University Computer Centre, with communication with the University Teaching Hospital (UTH) being through the conventional *FidoNet* system outlined above. Interfacing software to automate the transition between the two systems is being worked on elsewhere. Further "points" have recently been added at the Ministry of Health and the World Health Organization (WHO) Office in Lusaka.

Zambia was the first country in Africa to get an official licence from the PTT to operate the system on non-amateur frequencies and trials have proved successful. The system was recently inaugurated with an exchange of messages between the Queen, as Head of the Commonwealth in the UK, and the President of Zambia. The Queen was, at the time, visiting the University of Surrey where the microsatellites are made.

Within Africa, the system will prove particularly useful in exchanging messages with areas poorly served by conventional telecommunications. Messages have so far been exchanged with Mozambique, Kenya, Tanzania, and the Congo, as well as the USA and UK. The installation in the Congo is based at the African Headquarters of the WHO, and communication to local WHO branches is seen as being one major benefit of the system.

Likewise medical literature is to be made available via the satellite. The first *HealthNet Journal* has now been received (containing appropriate articles on developmental medicine) and search facilities will be added. A gateway with the Internet is also planned, through St. John's Memorial University in Newfoundland.

Once *HealthNet* is working successfully in Lusaka it is intended to expand the network to other areas in the country. Plans are being drawn up for e-mail provision for Provincial and ultimately District Hospitals in the rural areas. Communication will cover clinical consultation; distance learning; provision of literature; control of drug supplies; dissemination of epidemiological information; and management control.

ZangoNet

A group of leading Zambian non-governmental organisations (NGOs) has formed a steering committee (with consultants from the University) aimed at introducing a prototype e-mail/conferencing system within Lusaka in June 1992. It is intended that the NGO system will run as a cooperative along the same lines as Mango in Zimbabwe with which it is likely to develop close links.

Again the structure of the network will be *Fido*-based with its major international connections (for the benefit of sister/parent organisations outside Zambia) being thorough Mango and *GreenNet*.

ZangoNet is to be introduced in two phases. The first will enable ten NGOs with existing PCs to be provided with modems to allow them to connect to a host machine to be housed at the *Zambia Association for Research and Development* (ZARD) who have agreed to run the "hub" on a 24 hour basis. Phase two will provide fifteen of the major NGOs with no current microcomputer facilities with a PC and relevant connections.

Funding for Phase one has been somewhat delayed due to unexpected difficulties in obtaining donor funding. However funding prospects for the central PC and modems looked relatively optimistic at the time of writing (April 1992) and some equipment was already in place.

Modems were expected to be available at the 10 sites before June, by which time training would have been provided to the operators of the system at each participating NGO courtesy of NGONET.

The NGOs in Phase one have been selected so that they have a degree of commonality (i.e., a need to communicate) and represent the major groupings (e.g., health, green issues, gender issues, etc.) from among the 80 known NGOs in Zambia.

Once Phase two of the project starts, by late-1992, it is intended that organisations or offices outside Lusaka will be brought into the system. Before that time full experimentation on possible connection lines will need to take place.

Existing communication within the country is slow (a letter taking one week to be delivered within Lusaka, and up to three weeks elsewhere in the country) and there is much duplication of work that arises because of lack of knowledge of what others are doing. It is hoped that many problems will be redressed once *ZangoNet* is in place.

Conclusion

There are grounds for believing that electronic mail could become influential in improving information sharing within a country whose economy is undergoing a period of considerable strain.

It is appreciated that under-development is partly definable in terms of lack of access to information, in a world where information and power are closely equatable in both economic and political terms. The fact that some 90% of data on Africa is thought to be held on databases in the "West" is but one example.

Lack of access to foreign exchange (required for purchase of PCs and associated equipment) will clearly slow the process down: Unza is "doing e-mail on a shoestring." But it is believed that what is being experimented with is an appropriate technology; that it is sustainable; and that if Zambia is to maintain its place in the academic and economic worlds it must continue, with all the available help, to pursue this experimentation.

References

[1] Mike Lawrie, "Research and Academic Networking in South Africa," *ConneXions,* Volume 5, No. 8, August 1991.

[2] Steve Neighorn, Randy Bush, and Jeff Beadles, "Profile: RAINet," *ConneXions,* Volume 6, No. 5, May 1992.

MARK BENNETT holds a B.Sc. in Computing and Cybernetics from Reading University. After graduation he worked for 10 years in commercial computing in the UK before moving to Zambia as Systems Manager, and subsequently the Director of the University of Zambia Computer Centre. He is a Member of the British Computer Society and a Chartered Engineer. He can be reached via e-mail as: `Mark.Bennett@f1.n761.z5.fidonet.org`.

MIME: Multi-Media, Multi-Lingual Extensions for RFC 822 Based Electronic Mail

by Greg Vaudreuil,
Corporation for National Research Initiatives

Introduction

Request For Comments (RFC) 822 defines a commonly implemented message format for electronic mail. This message format, or close relatives to it, are used in Internet mail, USENET, BITNET and many other messaging systems. RFC 822 messages are in wide use in part due to their ease on implementation, and the increasing popularity of the Internet Suite of Protocols ("TCP/IP").

RFC 822 was written in 1982 for an environment where US ASCII text was the predominant communication form. Electronic messaging using RFC 822 has expanded well beyond this environment, to include both non-ASCII text and multi-media attachments. The *Multi-Media Internet Mail Extensions* (MIME) standard was developed to expand the RFC 822 message format to these new environments.

MIME

MIME is an extension to the RFC 822 framework both to enable the standard use of non-text body parts and to provide for the use of RFC 822 messages for non-ASCII languages. MIME is defined in two documents, the base document "MIME: Mechanisms for Specifying and Describing the Format of Internet Message Bodies" (RFC 1341) by Nathaniel Borenstein and Ned Freed, and a companion document "Representation of Non-ASCII Text in Internet Message Headers" (RFC 1342) by Keith Moore. MIME is the product of the IETF Internet Message Extensions Working Group.

At its most basic level, MIME defines a mechanism for declaring the content of a message and the transformations required to transport the message as if it were simple text. It extends e-mail capabilities into current environments by defining two new header fields, *Content-Type:* and *Content-Transfer-Encoding:*

The constraints

Extending the RFC 822 Message format in a backward compatible manner was a challenging endeavor in the diverse environment in which it is currently used. RFC 822 is used *very* widely, and is transported on many message transport services including SMTP, BITNET, UUCP (USENET protocols), NNTP (Network News), and commercial carriers. These transport environments impose varying restrictions on the message format beyond RFC 822. The reading and sending of RFC 822 is supported on most computer operating systems with a wide variety of local storage formats. These messages are written and read by many mail readers and composers with varying degrees of conformity to current protocols, and munged and altered by an assortment of more or less standard conforming software.

E-mail crossing the various connected networks (the collection of which is sometimes called "The Matrix" [12]) is altered and "munged." The following are some of alterations a message may be subjected to:

- The character set of the transport may be different, and the mapping to and from may not be reversible.
- Carriage return and linefeed may be deleted, added, or remapped.
- Lines may be truncated or wrapped anywhere over 76 characters.
- Message headers may be re-written, stripped, or reordered.
- Messages over 16K may be truncated.

Backward compatibility

Backward compatibility was one of the top priorities in the development of MIME. In the effort to be backward compatible, MIME not only extends RFC 822 in a backward-compatible fashion, but is written to work with existing software and practices to the maximum degree possible. While it is not possible to write new functionality into an existing protocol in such a way that current software will be able to display enhanced messages, it is the intent that a message written in MIME will be largely comprehensible to users of non-MIME software.

Content-Transfer encoding

To insure reliable delivery across the largest range of environments, two encoding schemes have been defined to convert the message contents into a form suitable for transport. *Base 64* was originally defined in RFC 1113 for use in sending Privacy Enhanced Mail. It is optimized for sending arbitrary binary data in a 7 bit transport environment. *Quoted Printable* was defined to encode data which has many of the characteristics of US-ASCII text but which may have long lines or some 8 bit characters. Quoted Printable has the advantage that encoded text generally remains human readable for users on an old style mail reader.

Three values for "no encoding" have been defined. When a message is 7 bit ASCII data, i.e., for all RFC 822 conforming mail today, an encoding type of *7bit* is used. For sending or storing 8 bit or binary data in an environment where suitable transport is available, e.g., a local file system, the Content-Types of *8bit* and *binary* have been defined.

Content-Type

The *Content-Type:* header field labels the information contained in the body of the message. This header is based on the RFC 1049 content-type, but extends the syntax and defines a hierarchical structure for the content-types. Eight top level Content-Types have been defined to give "hints" to gateways and user agents that do not recognize the specific Content-Type declared. The top level Content-Types are:

- *Text:* No special software is required to get the full meaning of the text aside from support for the indicated character set.
- *Multipart:* A body type which consists of multiple body parts, each of independent data types. This body part may be used recursively and is used to give structure to a multi-media document.
- *Message:* A body part which is itself a fully formatted RFC 822 conforming message which may contain its own different content-type. This type is also used for transparent fragmentation of large messages and used to provide "network pointers" to information available via other protocols such as FTP.
- *Application:* Raw data, typically uninterpreted binary data, or information to be processed by a mail-enabled application.
- *Image:* Data which requires a graphical display device such as a graphics terminal, printer, or fax machine to view the information.
- *Audio:* Data which requires an audio output device.
- *Video:* A body part which requires the capability to display moving images, typically including specialized hardware and software.
- *X-private:* Private Content-Types defined by bilateral agreement.

An initial set of Content-Types are defined in the base MIME document. These initial types reflect much of the current usage in the Internet. The addition of new content-types is accomplished by registering the content subtype name and a description of that type with the *Internet Assigned Numbers Authority* (IANA): `iana@isi.edu`.

continued on next page

MIME *(continued)*

The text Content-Type

MIME facilitates the use of non-ASCII character set needed to correctly represent non-English languages. The *text* Content-Type defines a character set (*charset*) parameter for identifying the character set of the text message. Because there does not yet exist a widely available universal character set, MIME provides a mechanism for declaring the character set in use. While interoperability problems will continue to exist between communities using different character sets, the ability to recognize the character set will allow them to communicate. It is expected that specific communities will document and register with IANA the character sets which are intended to be used.

Enhanced text format

The ability to annotate text with style information has proven to be quite popular on proprietary systems that support it in e-mail. On systems not supporting enhanced text, the functionality is expressed with varying forms of popular markings including *Asterix* for bold, lines of dashes for underlines, and the ever popular :–) smiley face.

MIME defines a simple mark-up for expressing many of these common functions, richtext. Among the supported formats supported by MIME are underlines, **bold**, *italics,* "quotations," and big/small type.

The Multipart Content-Type

RFC 934 and RFC 1154 defined experimental mechanisms for enclosing multiple body parts in the same message. Neither mechanism proved to be robust enough in the Internet mail environment. MIME defines a scheme based on RFC 934 to use a *boundary marker* to separate body parts, but unlike RFC 934, MIME requires that this boundary marker be declared in the multi-part body content-type header. Declaring the marker in each multi-part Content-Type allows each multipart content-type to be independent and therefore able to be nested indefinitely. The multipart Content-Type can be used for adding structure to a message. Both serial and parallel are defined subtypes. Message digests are provided for as a special case of multi-part.

Multi-lingual headers

The MIME specification limits itself to the body of an RFC 822 message. To facilitate the representation of non-ASCII information in the header, a companion document has been written to define a mechanism for encoding this important information like a personal name in the text portions of the message headers.

Defining a mechanism for representing non-ASCII data in the headers of messages is the most technically difficult problem in creating the multi-lingual standards. The solution space is limited by the common practice and requirement to have machine parsable headers. Headers are more frequently rewritten by mail transport agents than the message bodies. Headers are commonly deleted, rewritten, and reordered.

The chosen solution was to define an "encoded word" which would be interpreted by mail reading and transport software as an atomic unit. This unit would then survive as well as any other text in the reordering or reformatting of the header lines. This encoded word specifies the character set, the transport encoding, and the character data. This encoded word can be used anywhere in the headers where textual information is permitted including the To:, From:, Cc:, and Subject: headers.

Implementations

MIME is currently being implemented and MIME mail can already be seen on the Internet! At this writing, there are three openly-available implementations in use, and many commercial software providers are working on products.

References

[1] Crocker, D., "Standard for the format of ARPA Internet Text Messages," RFC 822, August 1982.

[2] Rose, M. T. and Stefferud, E. A., "Proposed standard for message encapsulation," RFC 934, January 1985.

[3] Robinson, D. and Ullmann, R., "Encoding header field for internet messages," RFC 1154, April 1990.

[4] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1341, June 1992.

[5] Moore, K., "Representation of Non-ASCII Text in Internet Message Headers," RFC 1342, June 1992.

[6] Borenstein, N., "A User Agent Configuration Mechanism For Multimedia Mail Format Information," RFC 1343, June 1992.

[7] Borenstein, N., "Implications of MIME for Internet Mail Gateways," RFC 1344, June 1992.

[8] Simonsen, K. "Character Mnemonics & Character Sets," RFC 1345, June 1992.

[9] Borenstein, Nathaniel S., "Multimedia Mail From the Bottom Up—or Teaching Dumb Mailers to Sing," *ConneXions,* Volume 5, No. 11, November 1991.

[10] Borenstein, Nathaniel S., "*Metamail*—Multimedia Mail for the Masses," *ConneXions,* Volume 6, No. 3, March 1992.

[11] Rose, M. T., *The Internet Message—Closing the Book with Electronic Mail,* ISBN 0-13-092941-7 (book to be published by Prentice Hall, October 1992.)

[12] Quarterman, John S., *The Matrix: Computer Networks and Conferencing Systems Worldwide*, Digital Press, ISBN 1-55558-033-5, 1990.

[13] Quarterman, John S., "Mail Through the Matrix," *ConneXions* Volume 3, No. 2, February 1989.

[14] Sirbu, M. A., "Content-type header field for Internet messages, RFC 1049, March 1988.

GREG VAUDREUIL serves as the Chairman of the IETF Internet Message Extensions Working Group overseeing the development of MIME and serves as the IESG Secretary for the IETF Secretariat at the Corporation for National Research Initiatives. He graduated from Duke University with a degree in Electrical Engineering and a major in Public Policy Studies. He can be reached via e-mail as: `gvaudre@NRI.Reston.VA.US`.

An Analysis and Comparison of Internet and X.400 Messaging

by Daniel Blum, Rapport Communication

Introduction

With the advent of a new set of *Multipurpose Internet Mail Extensions* (or MIME) [1] standards, the Internet RFC 822 [2] and *Simple Mail Transfer Protocol* (SMTP) [3] mail infrastructure is poised to take a leap forward into the multimedia messaging paradigm of the 1990s. Because MIME will have a major impact in the Internet and beyond, it is important to compare Internet Mail standards in general and MIME in particular with the 1988 X.400 and X.500 global messaging/directory standards and to consider how they will evolve and interwork in the global messaging environment.

The SMTP and RFC 822 Internet Mail standards are products of the 1980s and had long lain dormant. Their reawakening was originally stimulated by the emergence of ambitious X.400 messaging [4, 5] and X.500 directory [6] standards in 1984 and 1988 as well as by Internet user demand for more advanced messaging facilities such as convenient binary file transfer. Initially, the hope was that X.400 could satisfy advanced Internet messaging requirements; however, while X.400 has numerous adherents in the Internet, many others have significant cultural and logistical problems with the standard. Factors inhibiting X.400 acceptance include an addressing scheme that is very different from RFC 822's, the unavailability of electronic copies of the standard, the complexity of its implementation, and logistical difficulties with registration of X.400 address components.

Originally, dealing with X.400 in the Internet took the form of limited installation of the standard and more widespread usage of SMTP to X.400 gateways. MIME was born from a tacit understanding between the "pro-X.400" and the "anti-X.400" forces on the Internet that, as users began demanding multimedia capabilities well-attuned to their installed RFC 822-based mail systems, a more decisive solution than partial X.400 penetration and gateways would have to be found.

In the electronic messaging fabric of the 1990s, X.400 and Internet Mail will coexist alongside proprietary protocols (some of which, such as Novell's *Message Handling System* (MHS) and IBM's *System Network Architecture Distribution System* (SNADS)) are important *de facto* standards. To understand how X.400 and Internet Mail will coexist and where each will be used, we will begin by considering how each addresses the major functional elements of a comprehensive messaging solution. We will conclude by offering a dispassionate messaging industry assessment of the future roles of X.400 and Internet Mail in the Internet environment, the commercial backbone messaging environment, and the commercial LAN e-mail environment.

Anatomy of a Messaging Solution

In order to compare MIME, X.400, and (to a lesser extent) other mail standards, we must first diagram the anatomy of a total messaging solution. This anatomy has changed over time. Originally, mail system implementations sported simple text-based message formats and consisted of little more than a user interface and a transport capability. Mail storage and directory functions were rudimentary and deeply embedded in the user interface or transport functions.

Even as the standards have developed, products and user expectations have moved to keep pace. No longer are users satisfied with simple text-based proprietary mail systems. They now demand interoperability between mail systems and their respective directories.

They also demand products that support integrated messaging features such as binary attachments containing editable files, spreadsheets, or even multimedia (e.g., audio and video) message content objects.

Also in this brave new world of integrated messaging, mail enabled computer applications are sending one another mail to perform office automation duties; store-and-forward electronic data interchange (EDI) between companies can itself be viewed as no more than a large scale mail enabled application.

Product architectures have become more sophisticated, providing interoperable transport based on X.400 or other protocols, more flexible user and application program interfaces, more sophisticated electronic directories of users, and even sophisticated message storage and authorization facilities. Increasingly, these architectures allow users or implementors to "plug and play"—that is, one can utilize Product X's user interface with Product Y's directory, and Product Z's message store. Figure 1 displays the architecture of emerging top-of-the-line messaging facilities (composed of multiple products, access methods, and technologies).

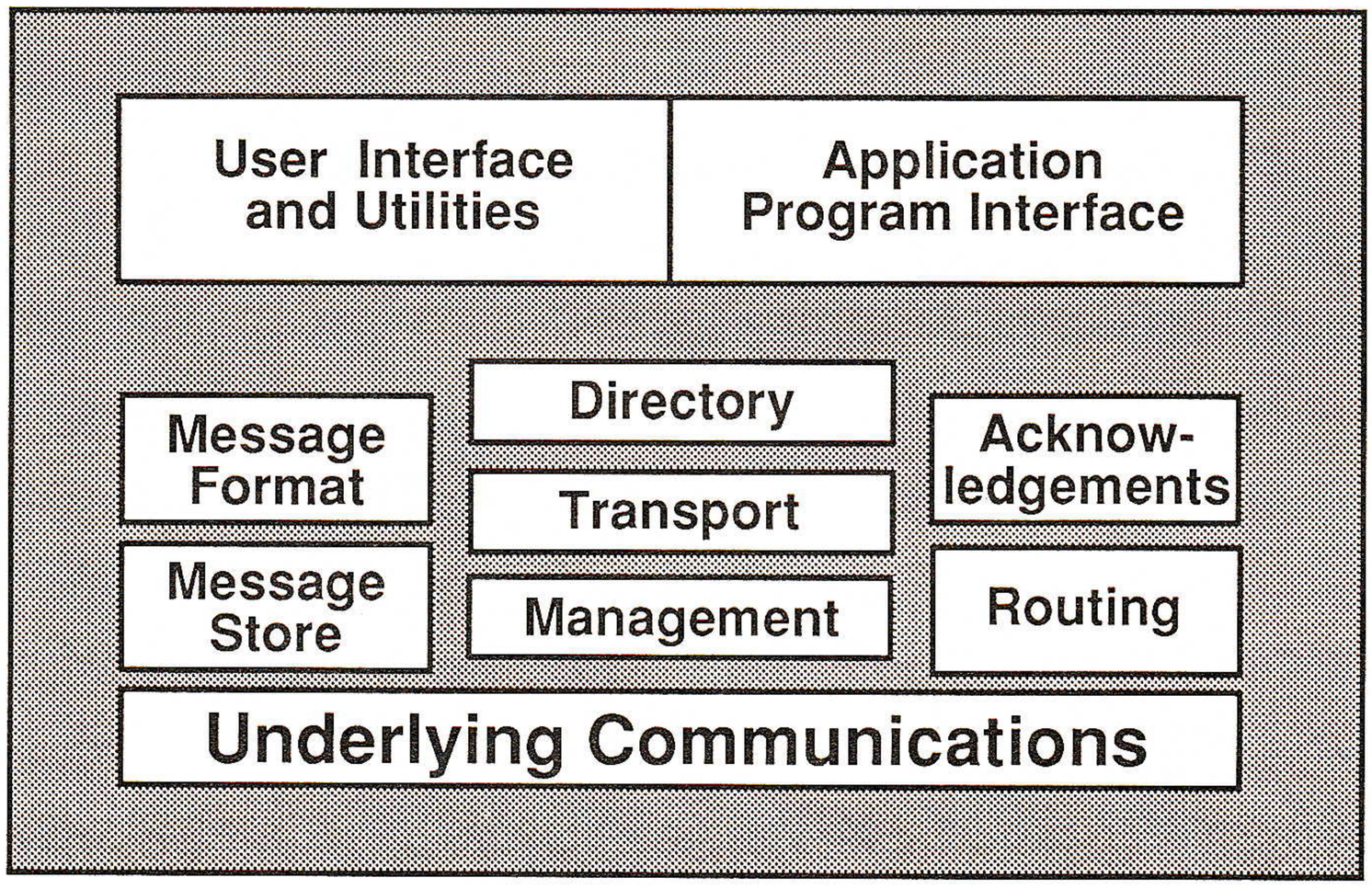

Figure 1: Elements of a Total Messaging Solution

Thus, a useful way to structure our comparison is to evaluate X.400 and Internet mail in terms of their ability to perform any or all of the functions from Figure 1.

Transport

Transport differences between Internet Mail and X.400 are one of our most important considerations. Under the transport category, we will contrast Internet Mail and X.400 message formats, acknowledgment, routing strategy, and underlying communications. Subsequently, we will cover directories, addressing, and other functions.

Message formats and header fields

At the highest level of message format, X.400 divides a message into an *Envelope* (called P1) and a *Content* (called P2 or P22 for 1988 X.400). The P22 content itself consists of a *Heading* and a *Body*. RFC 822 divides messages into a *Header* and a *Body* only.

Table 1 contrasts the X.400 P22 (1988) message heading with the RFC 822 message header. The heading/header fields shown in adjacent columns generally have a straightforward mapping that has been described in RFC 987 [7] and RFC 1148 [8].

continued on next page

Comparing Internet and X.400 Messaging *(continued)*

X.400 Fields	*RFC 822 Fields*
this-IPM	message id, resent message id
originator	from, resent-from
authorizing-users	—
primary-recipients	to, resent-to
copy-recipients	cc, resent-cc
blind-copy-recipients	bcc
replied-to-IPM	in-reply-to
obsoleted-IPMs	—
related-IPMs	references
subject	subject
expiry-time	—
reply-time	—
reply-recipients	reply-to
importance	—
sensitivity	—
auto-forwarded	—
extensions	extension-field, user-defined-field
incomplete-copy	—
language	—
—	orig-date, resent-date
—	keywords
—	comments

Table 1: Mapping of X.400 to RFC 822 Fields

In addition, MIME introduces a number of new header fields including: *MIME Version, Content-Type, Content Transfer Encoding, Content ID,* and *Content Description.* X.400 P1 and P22 have Content Type and Content ID fields, while the OSI Presentation Layer provides extensive mechanisms for negotiating the encoding of a message or any other type of application data.

The X.400 P1 envelope contains many additional fields that provide flags and control information for messaging services. A few of the more important services enabled by P1 envelope control information are shown in Table 2 below.

P1 Envelope-enabled Services:
originator specified priority
conversion
distribution list expansion
redirection
security (authentication, confidentiality, integrity, and non-repudiation)
requested delivery method
delivery report request

Table 2: Messaging Services Provided Through P1 Envelope

When we come to the P1 envelope and the area of backbone services, comparing X.400 with Internet Mail systems is rather like comparing a 747 airliner with the original Volkswagen Beetle. With MIME coming down the turnpike, however, the Internet Mail Volkswagen is about to grow wings.

Body parts

Taking MIME into account, let us leave the topic of message headers and move to a discussion of message body parts. (Since frequent sentences containing the word "body part" might make our discourse excessively scatological, we will use the terms "attachment" and "body part" interchangeably). We begin by considering the differences between a single X.400 message body part and a single MIME message body part. Here again, the standards have quite different approaches.

MIME uses a "type" and "subtype" mechanism where the base types are text, multipart, application, message, image, audio, and video. There are a number of standard subtypes defined in the draft MIME RFC. Implementations thus embed "type/subtype" text strings such as "text/richtext" or "image/g3fax" in messages to tag the appropriate attachments. Users or implementors can also interchange information tagged with private subtype values (beginning with the string "X–") and/or they can register new subtype values with the *Internet Assigned Numbers Authority* (IANA).

OID

The 1988 version of X.400 defines a list of standard body parts and then an external body part that can carry information labeled by an *Object Identifier* (OID) data value. The OID is a hierarchical number registration system defined in OSI's *Abstract Syntax Notation One* (ASN.1) standard. The scheme produces unambiguous numbering of objects for identification purposes while allowing registration authority to be delegated from higher to lower level groups. In the case of Internet objects the OID registration authority begins at ISO (1) and then passes to "identified organizations" (6), the DoD (3), and the Internet (1). Thus (in a somewhat simplistic example) if the Internet was to define an X.400 body part (20), the external body part's OID would be "1.6.1.3.20" (since all Internet OIDs begin with 1.6.1.3).

Users or implementors can register OID values or arcs in the OID tree in multiple places, including the American National Standards Institute (ANSI) and the IANA. Registering object identifiers with ANSI incurs a cost of $2,500; for this reason, commercial organizations and vendors often buy one "branch" from ANSI and then register OIDs for free beneath that branch. Note that companies have no legal obligation to use ANSI and may choose to derive their registration authority elsewhere. Thus, both MIME and X.400 use various predefined types and each possesses an extension mechanism. X.400 and MIME body parts are compared in Table 3 below.

X.400 P22 Body Part	*MIME Body Part*
ia5-text	*text* type with *plain* and *richtext* subtypes and optional use of character sets
voice	*audio* type and subtypes
g3-facsimile	*image* type, *g3fax* subtype
g4-class1	image type, new subtype could be registered
teletex	*text* type, using character set selection mechanism
videotex	could be implemented as a MIME *application* type
encrypted	could be implemented as a MIME *application* type
message	*message* type
mixed-mode	could be implemented as a MIME *application* type
bilaterally-defined	within any MIME type, a private subtype
nationally-defined	within any MIME type, a "nationally defined" subtype
externally-defined	equivalent to any MIME type/subtype

Table 3: MIME and X.400 Body Parts

continued on next page

Comparing Internet and X.400 Messaging *(continued)*

Mechanisms

It should be clear that MIME and 1988 X.400 provide different mechanisms for labeling and sending virtually any type of data. X400 can send the equivalent of MIME's video type using its externally defined body part. MIME's type/subtype mechanism can represent any externally defined or standard X.400 body part. One advantage of MIME is as follows: since all MIME applications or mail *user agents* (UAs) will recognize each of MIME's short and broadly known list of types, a MIME UA always starts out with some information about an unknown body part. This knowledge could assist the MIME UA in making intelligent decisions; e.g., it is appropriate to display an unknown type of text but not audio!

It is in the handling of multipart bodies and encapsulated messages that the architects of MIME displayed considerable creativity. Whereas X.400 simply allows a message to contain multiple body parts in an unstructured list and also allows for the encapsulation of messages during forwarding, MIME allows for the originator to do the following:

- Instruct the receiving UA to render attachments sequentially or in parallel (e.g., audio and video).
- Send attachments as alternatives to one other (e.g., same letter in text and richtext). UAs choose whichever of the alternative body parts is the most appropriate for rendition.
- Encapsulate partial messages as a content type, allowing big messages to be sent in fragmented form. This feature is useful for interworking with geriatric mail gateways plagued by severe size limits. It could be even more useful in the future when large video clips will be sent as messages. On the other hand, X.400 has no theoretical size limit; in practice the NIST OSI Implementors Workshop Agreements Chapters 7 and 8 [9] mandate support for messages of at least 2 megabytes in length.
- Indicate that a message content is external; that is, the message content has not actually been sent. Instead, the message encloses instructions for FTP, Mail Server, or other methods of remote file access. Thus, recipients can ignore attachments containing uninteresting information or uninteresting alternatives (see first bullet above) without up front bandwidth wastage.
- Indicate within the individual body part what conversions (*compress, uncompress, tar,* etc.) need to be performed—this last feature only works on "application" type attachments.

While none of the above capabilities are present "out of the box" in X.400, all could be retrofitted into conformant X.400 implementations. The present draft of "Mapping MHS/RFC-822 Message Bodies" (Miles, Thompson, Rose) simply encapsulates the more exotic MIME objects in X.400 as the external X.400 body part "MIMEBodyPart." This forecasts the interesting hybrid possibility of a MIME User Agent on top of X.400 transport.

We should also note a few virtuoso performances from body part cloners in the X.400 standards community. One idea—which came out of the X.435 (EDI over X.400) standards [10] effort—is the *Cross Referencing Information Field* in the EDI Message Heading that allows applications to identify one or more body parts in the current message or in other messages and to cross reference them in some fashion.

Another piece de resistance is X.400's *File Transfer* body part, a data structure that can carry detailed information about files embedded in messages. While the File Transfer Body Part was originally developed for the 1992 X.400 standard, the NIST OSI Implementors Workshop and the Electronic Mail Association's PRMD Operators Group are actively encouraging X.400 vendors to retrofit it into their 1984 and 1988 X.400 implementations. Figure 2 shows an example of the File Transfer Body Part.

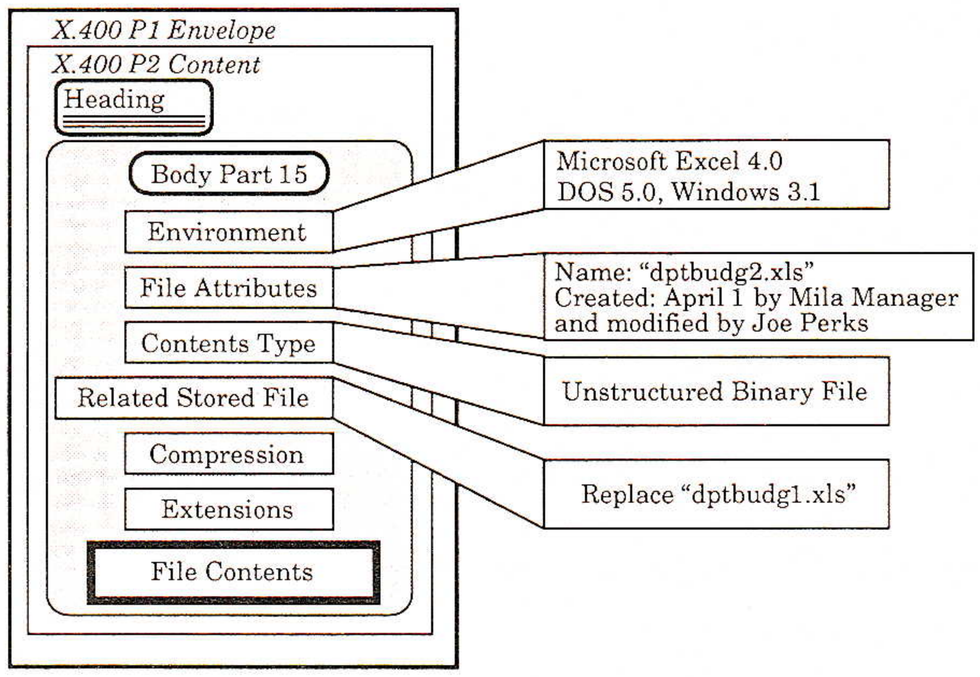

Figure 2: The X.400 File Transfer Body Part

Most of the fields in the file transfer body part are optional. They are, incidentally, compatible with OSI's *File Transfer, Access, and Management* (FTAM) standard. If the main intent of mail-based file transfer is just to be able to provide a way for the recipient's system to launch an application, MIME's type/subtype information should be sufficient, provided that enough application subtypes are registered to take account of platform and version diversity. However, X.400's file transfer body part provides a more comprehensive set of features.

Interworking between MIME and X.400

A draft RFC by Robert S. Miles, Steve Thompson, and Marshall T. Rose [11] extends RFCs 987 and 1148 to provide for mapping between X.400 message formats and MIME message formats. The draft RFC's general approach is to map built-in MIME Body Parts to their X.400 counterparts (e.g., it maps a "image/g3fax" body part into a *IPMS.G3FacsimileBodyPart*) and vice versa. When translating to X.400 where there is no X.400 counterpart to a MIME Body Part (e.g., "multipart/parallel"), the draft RFC calls for encapsulating the entire MIME Body Part as an X.400 Body Part labeled with a special "MimeBodyPart" OID value. In the other direction where there is no MIME counterpart to an X.400 body part, the draft RFC calls for encapsulating the attachment and labeling it with the string "application/oid" followed by its actual OID.

The Internet Assigned Numbers Authority (IANA) will maintain a table of correspondence between the X.400 object identifiers and MIME character oriented type/subtype definitions. There is as yet no mapping facility defined for translating the X.400 File Transfer body part to MIME and vice versa other than the default method of encapsulation.

Acknowledgements

As users we sometimes like to know if our electronic mail was delivered, and, if delivered, was it read. This requires some form of automatic acknowledgment message from the recipient or the recipient's message transfer agent (MTA) to the sender. Internet Mail currently does not have a standardized acknowledgment or error message format, although most systems utilize a Postmaster mailbox to generate automatic non-receipt notices.

continued on next page

Comparing Internet and X.400 Messaging *(continued)*

This works well enough when the mail gets as far as the receiver's domain, however, the author has lost numerous Internet messages without a trace in gateways and suspects his experience is by no means unique. Today, an Internet group called "ietf-ack" is pondering the semantics of acknowledgments in general and the practical considerations of retrofitting such onto Internet Mail.

X.400, on the other hand, sports two types of acknowledgment messages: the delivery notification (sent when the MTA delivers to a mailbox or a gateway) and a receipt notification (sent either manually or automatically at the time the user reads the message or takes other definitive actions such as forwarding or discarding it). Messaging users can request delivery or non-delivery reports from the recipient's MTA. After expansion, MTAs may also dispatch delivery reports to the owner of an X.400 distribution list. E-mail users can request receipt or non-receipt notifications from the recipient's UA. Different X.400 Content Types define different types of notifications.

Routing

Internet Mail routing and X.400 routing presently operate in fundamentally different ways. From the point of view of an MTA, much Internet routing is performed at the source when the sending MTA obtains MX records from the *Domain Name System* (DNS) which (usually) identify the MTA at or near a recipient's end host. This works well from the point of view of both the MTA and the user.

X.400 serves many communities, not just one relatively cohesive Internet community. It does not have a DNS, thus routing must be performed incrementally at each MTA using *Originator/Recipient Addresses* (OR Addresses) within messages. This incremental routing has the advantage of allowing very different, loosely coupled, and distrustful communities with different views of addressing/routing to interwork. The downside, however, is that routing tables must be maintained manually at multiple sites.

Anecdotal evidence suggests that Internet Mail and X.400 Mail routing will become more alike than unlike as the years pass. Both CCITT/ISO standards makers and Internet X.400 adopters are planning to store X.400 routing information in the X.500 directory so that those communities who are willing to share this kind of information can interwork more directly and less incrementally. At the same time, Internet Mail routing will become more incremental as the community expands and continues diversifying—already many organizations use mail gateways as firewalls and are not well-connected to the DNS.

Underlying communications

Mail transport presupposes some form of underlying communications. By these considerations, Internet Mail is more tractable for two reasons. First, TCP/IP has flourished while OSI networks have been slow to take hold. Secondly, RFC 822 and MIME are deliberately written to be independent of the transport mechanism and messages in these formats are commonly sent over dialup lines using UUCP as well as SMTP. There is a rich body of folklore and practical experience on the Internet that ensures quite acceptable levels of interoperability and flexibility to boot.

If one considers the OSI reference model in the most abstract sense, X.400 is also supposed to be independent of underlying communications. However, with the exception of the RFC 1006 *de facto* standard for running X.400 over TCP/IP, the disjoint communities of X.400 have not yet been able to standardize or profile means of transport other than OSI connection oriented or connectionless networking.

Some X.400 vendors have X.400 running over dialup lines; however, insofar as we know, none of the different implementations can yet interwork in dialup mode. This and the memory demands of the OSI protocols have hindered the adoption of Native X.400 UAs.

Addressing and Directory

Both the Internet and X.400 define a global addressing scheme. However, there are three major differences between the schemes:

- The Internet scheme is oriented toward the private mail domain community, whereas the X.400 addressing scheme is oriented towards both public and private mail. In practice, public messaging service providers—called *Administration Management Domains* (ADMDs)—have dominated the allocation of X.400 domain names in a manner that has been detrimental to usage.

- Registration of mail domain names in the Internet is cheap, easy, and supported by an automated Domain Name System (DNS) that allows any MTA to map a domain name to a host address. By contrast, *Private Management Domain* (PRMD) name registration for X.400 does not yet exist at the national level and there is no guarantee that two different service providers won't allocate duplicate mail domain names to their customers. However, X.400 PRMD name registration exists within the Internet and agreement (through a subcommittee of the U.S. CCITT Study Group D) is pending on a national PRMD name registry.

- Internet addresses are fairly easy to type in and follow a consistent formula. For example, the author's Internet mailbox is: "`4108980@mcimail.com`." The corresponding X.400 mailbox is "`C=US ; ADMD=MCI ; DDA=ID=4108980`." Having to type in coded fields (through some user interfaces at least) and the lack of a canonical printed representation of the X.400 O/R Address have been stumbling blocks for many users.

Electronic mail directories are the industry's riposte to the inconvenience of having to remember or guess e-mail addresses—be they simple or inscrutable in form. Happily, directories are one area where the Internet and the international standards community largely agree. Efforts to develop X.500 Directory Service operational and interconnection experience are underway in the Internet through the Internet *White Pages Pilot* and in the X.400 ADMD community through the North American Directory Forum. The time may come when both X.400 OR addresses and RFC 822 addresses will live in the same global, distributed, electronic directory.

As one might expect, given the standards-community origins of X.500, X.400 message handling standards leverage the directory to a greater extent at this stage than do their Internet counterparts. Both X.400 and Privacy Enhanced Mail UAs or MTAs can utilize the directory to store unforgeable public encryption keys (called *Certificates*). However, only X.400 at the present time uses X.500 to perform capabilities assessment (e.g., to answer questions such as: "can user X accept an attachment of Format Y?" and/or "what is user X's preferred delivery method?"). Only X.400 uses the X.500 directory as the repository for shared distribution lists that are expanded at the MTA level subject to stored "submit permissions."

Application Program Interfaces

Somewhere in the anatomy of a messaging solution, it is desirable that there be an open application program interface. From that exposed interface, applications can access message transport services.

continued on next page

Comparing Internet and X.400 Messaging *(continued)*

The mapping between an exposed API and a transport service need not be one to one; for instance, the same API could provide access to Internet Mail, X.400 mail, or other transports. However, an API which is "native to" or written expressly for a transport service will generally provide better or fuller access to the features of that transport service.

Neither the Internet Mail standards nor the X.400 international standards address APIs. With mail enabled applications proliferating, however, it is no surprise that three major de facto standard APIs are vying for dominance of the messaging industry.

The first API of significance is one created by a group of vendors that banded together in the late 1980s to form the X.400 API Associations. The X.400 API [13] has since been implemented or is in the process of being implemented by a number of major vendors and currently supplies the glue between at least a half dozen multiple vendor MTA to gateway/UA product pairings. This API specifies **C** programming interfaces by which messages can be submitted and delivered to UAs or MTAs. It has been extended to incorporate an X.500 Directory Services API, an X.435 (EDI) API, and an X.400 Message Store API. X/Open has joined the X.400 API Association in endorsing the API and the API is currently under consideration by the IEEE for inclusion in POSIX.

However, the X.400 API was never designed for casual application programmer usage and is really more appropriate for system developers. For this reason, it is not universally accepted among vendors and most vendors provide higher level "toolkits" or subroutine interfaces. Recently, there has been a spate of attempts in the commercial LAN e-mail environment to establish vendor APIs as de facto standards. Currently, the most important such APIs are the *Microsoft Messaging API* (MAPI) and the self-termed *Vendor Independent Messaging* (VIM) API [14] from a group of major LAN desktop software and system vendors.

Message Store

Internet Mail does not as yet have a recognizable protocol or processing entity dedicated to message storage. By contrast, the 1988 version of X.400 defines a *Message Store* entity. User Agents can access the X.400 Message Store via a protocol called P7. The Message Store (MS) also submits messages to, and accepts deliveries from, X.400 MTAs using either X.400's P3 protocol or (more often) co-located application program interfaces. In addition to database style retrieval operations on stored messages, the MS offers hooks for the definition of automatic actions to be taken upon messages (such as forwarding, deleting, or user-defined activities).

The X.400 solution has gained favor among many public service providers, specialized service providers, and even corporate/government MIS service providers who would like to support a single message retrieval interface. A few 1988 MS implementations are now available. However, in the commercial LAN e-mail market sector, X.400 Message Stores with their OSI-based P7 protocol and complexity of implementation have yet to gain significant market share.

Nevertheless, the concept of storage being a separable part of messaging architecture is gaining favor in the marketplace. LAN clients and workstations need shared storage on LAN servers.

Mail enabled applications not only need storage but also rules-based automatic actions. The X.400 and VIM APIs as well as other APIs from major vendors all contain message store API elements and, increasingly, products are being built to plug and play.

Security

X.400 and Internet Mail have both developed secure messaging capabilities. Internet Mail's security capabilities were developed as part of the *Privacy Enhanced Mail* (PEM) effort [15]. A partial list of security services provided by the 1988 version of X.400 is given below:

- *Authentication* of message, delivery report, or probe origin and/or the identity of a UA, MTA, or Message Store at connection establishment time.
- *Integrity* (through digital signatures) of messages, delivery reports, and probes.
- *Confidentiality* (through encryption) of message contents.
- *Non-repudiation* of submission or delivery of a message.
- *Other services* such as message sequence integrity, message security labeling, and secure access management.

The NIST OSI Implementors Workshop sorted these capabilities into six classes: S0, S0a, S1, S1a, S2, and S2a. The "a" in S0a, S1a, and S2a denotes a confidentiality capability—thus S0 and S0a are the same except that S0a offers encryption. S0 and S0a provide UA-oriented end to end security services such as message origin authentication, content integrity, content confidentiality, and non- repudiation of delivery. S1 and S2 provide MTA to MTA security services such as non-repudiation of submission, peer entity authentication, etc.

PEM, on the other hand, is basically a variation on the RFC 822 message body type that can reside within an RFC 822 header. Its major capabilities comprise the equivalent of X.400's content integrity service using signatures and content confidentiality service using encryption. It does not have the rich variety of services present in 1988 X.400, but on the other hand it probably has the services the majority of security-conscious users are likely to demand—the only major omission seems to be non-repudiation of delivery.

Interworking considerations require procedures to map PEM to MIME and PEM to X.400. In the first instance, PEM to MIME mapping will eventually need to take the form of carrying a MIME body part as a PEM body part and of carrying PEM body parts within MIME. In the second instance, X.400 to PEM interworking will be a "tunnelling-style" solution—the NIST OSI Implementors Workshop already recommends that entire PEM messages be encapsulated as X.400 body parts.

Conclusion

We have compared the feature sets of Internet Mail and X.400/X.500 messaging standards, discussed interworking, and tried to project how well each standards set is positioned to provide a total messaging solution (as was diagramed in Figure 1 above). At this point, it is appropriate to peer deeply into our crystal ball and consider the future of X.400 and Internet messaging in the Internet environment, in the commercial backbone messaging environment, and in the commercial LAN e-mail / workflow environments.

Internet Mail as it presently exists and supplemented by MIME is likely to dominate the Internet community because the technology is custom-fit to the installed base and is relatively easy to implement. X.400 will still be used on the Internet but not as widely.

continued on next page

Comparing Internet and X.400 Messaging *(continued)*

X.400, on the other hand, will dominate in the commercial backbone messaging environment where it is already both the accepted and the de facto means of communication between global public messaging services, is internationally adopted as a required element of government procurements, and is strongly favored as a backbone by those Fortune 500 corporations with a heterogeneous installed base of legacy mail systems. Because of these factors, all major system vendors support some kind of X.400 product or gateway if for no other reason than to avoid exclusion from major information system procurements. Moreover, a number of vendors are making X.400 and X.500 backbone provision into a profitable business in its own right. Products based on the 1988 standards are emerging and their price-performance has and will continue to improve dramatically [16].

Some might ask, what about a Commercial Internet? Could it provide an alternative commercial backbone messaging solution in which Internet Mail displaces X.400? At present this seems unlikely. The Commercial Internet is still only a dream, and the MIME process has yet to produce robust messaging backbone machinery such as message acknowledgments and message conversions. While Internet Mail as a whole is vendor-independent, it is not community independent in the same sense as X.400. MIME has been carefully crafted to be backward compatible with conformant and even non-conformant SMTP/RFC 822 implementations. Many of these implementations are not capable of 8-bit data transport. In making the compromises necessary to accommodate the Internet installed base, MIME and Internet Mail will not necessarily emerge with an any-to-any backbone protocol.

All the same, the fact that X.400 is likely to retain its dominance in the commercial backbone messaging environment should not encourage X.400 advocates to rest on their laurels. There remain significant problems with X.400 that must be solved if X.400-based commercial backbone messaging is to achieve its ideal of providing ubiquitous integrated messaging connectivity for E-mail, EDI, and all manner of applications. Problems that are having an immediate short term impact are the lack of 1) an international registration system for either external body parts or for private domain names, 2) a P2 downgrading mechanism that will convert 1988 X.400 messages with external body parts to 1984 X.400 messages with unidentified body parts, and 3) a widely accepted de facto standard for running X.400 messages over asynchronous telephone connections. All of these issues are being worked in various fora and need to be solved sooner rather than later. Longer term, global X.500 interworking must be put in place and standards will be needed for storing routing tables in the directory.

Having discussed the Internet messaging and commercial backbone messaging environments, there remains the commercial LAN e-mail/workflow environment—or the real prize. It turns out that the commercial LAN e-mail market is growing more rapidly than the Internet, more rapidly than X.400, and certainly more rapidly than the gross national product. In this environment, neither X.400 mail nor Internet Mail appear likely to dominate—instead, products from the major LAN systems and desktop software vendors will take the lion's share while leaving plenty of lucrative niches for products implementing innovative workflow and other solutions ahead of their time.

However, LAN e-mail users will be able to use or access either X.400 or Internet Mail indirectly. Most LAN e-mail implementations have both SMTP and X.400 gateways and they will continue to improve on these as the technology evolves.

Nor will the choice be restricted to gateways. Native X.400 UA implementations seem to be enjoying some success recently with one vendor claiming to have picked up 100,000 users in about 6 months. This market, largely composed of North American and European government customers and Fortune 500 companies will continue to grow. At the same time, the commercial LAN e-mail world will grow ever more hospitable to the Internet Mail protocols as DOS, UNIX, and the major network operating systems converge. Native MIME UAs will make a debut, probably a significant one.

If we choose to look at this new bipolar messaging problem from the "Holy War" perspective, Internet Mail's greatest advantage is that it has a registration and routing infrastructure in place. X.400's greatest advantage is that it has a full features set crafted to offer industrial strength, commercial grade, any-to-any store-and-forward communication. If these standards are going to battle for global messaging dominance, Internet Mail could be viewed as an infrastructure in search of features and X.400 could be viewed as a features set in search of an infrastructure. The standard that can put all the pieces together first will "win." Since we are currently in the bottom of the first inning and the author chooses only to forecast as far as approximately the third inning (where the score will be tied at "1–1"), readers are encouraged to keep their own scorecards and stay tuned for further announcements.

If we adopt a "coexistence and hybridization model," we could argue that messaging users will be the better for having both Internet Mail and X.400, and other protocol choices because, however imperfectly, they will be able to exchange "strongly typed" message attachments sooner between their respective communities. Users will also have increasingly comprehensive directory capabilities provided through X.500 and other means of directory exchange or synchronization. Taken together, directories and attachments will provide much of the essential infrastructure for mail enabled and workflow applications—also needed are (at a minimum) genuinely vendor independent APIs, vendor independent forms, a standard language for executable messages, and standard electronic document authorization techniques.

In this multiprotocol messaging environment, a great deal more thought needs to go into how MIME, X.400, and other protocols will interwork. Messaging gateway technology definition has so far been left to individual vendors and is poorly understood by the messaging community in general. Integrated global messaging will arrive, but its birth can be made much less painful if "Internet bigots," "X.400 bigots," and "LAN e-mail bigots" bury the hatchet, develop common terms of reference for backbones, gateways, directories and APIs, and let hybridization happen.

References

[1] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1341, June 1992.

[2] Crocker, D., "Standard for the format of ARPA Internet Text Messages," RFC 822, August 1982.

[3] "Simple Mail Transfer Protocol (SMTP)," RFC 821.

[4] CCITT, "Data Communications Networks Message Handling Systems Recommendations X.400–X.430," 1984.

continued on next page

Comparing Internet and X.400 Messaging *(continued)*

[5] CCITT/ISO, "Data Communications Networks Message Handling Systems Recommendations X.400–X.420," aka ISO 10021, 1988.

[6] CCITT/ISO, "Data Communications Networks Directory Services Recommendations X.500-X.521," also known as ISO 9594, 1988.

[7] S.E. Kille, "Mapping between X.400 and RFC 822," RFC 987, June 1986.

[8] S.E. Kille, "Mapping between X.400 (1988) / ISO 10021 and RFC 822," RFC 1148, March 1990.

[9] "Stable Implementation Agreements for Open Systems Interconnection Protocols Version 5 Edition 1," December 1991.

[10] CCITT, "Message Handling: EDI Messaging Service (F.435)" and "Message Handling System: EDI Messaging System (X.435)," 1991.

[11] Robert S. Miles, Steve Thompson, and Marshall T. Rose, "Mapping MHS/RFC-822 Message Bodies," Draft RFC, March 1992.

[12] Marshall T. Rose, and Dwight E. Cass, "OSI Transport Services on top of the TCP," RFC 1006, May 1987.

[13] X.400 API Association, X/Open Company Ltd., "X.400 API Specification, Version 2.0 (XAPIA)," 1989–1991.

[14] Apple, Borland, Lotus, and Novell Corporations, "Vendor Independent Messaging Implementation Guidelines," March 1992.

[15] J. Linn and S. Kent, "Privacy Enhancement for Internet Electronic Mail: Parts I, II, and III," RFC 1113, 1114, 1115, August 1989.

[16] Daniel J. Blum and DML Associates, "The Messaging Connection: Integrated Messaging and Directories Based on X.400/X.500 Standards in the 1990s."

[17] Borenstein, Nathaniel S., "Multimedia Mail From the Bottom Up—or Teaching Dumb Mailers to Sing," *ConneXions,* Volume 5, No. 11, November 1991.

[18] Borenstein, Nathaniel S., "*Metamail*—Multimedia Mail for the Masses," *ConneXions,* Volume 6, No. 3, March 1992.

[19] Rose, M. T., *The Internet Message—Closing the Book with Electronic Mail,* ISBN 0-13-092941-7 (book to be published by Prentice Hall, October 1992.)

[20] Quarterman, John S., *The Matrix: Computer Networks and Conferencing Systems Worldwide*, Digital Press, ISBN 1-55558-033-5, 1990.

[21] Quarterman, John S., "Mail Through the Matrix," *ConneXions* Volume 3, No. 2, February 1989.

[22] Sirbu, M. A., "Content-type header field for Internet messages, RFC 1049, March 1988.

DANIEL J. BLUM has been active in open networking research and development since the early 1980s. He is well-known as a writer, educator, technological forecaster, and consultant on integrated messaging and directory services. He has worked with key standards development committees, implemented communications systems, conducted extensive surveys of industry products, and advised industry and government users on the planning, architecture, procurement, and deployment of modern networks and value added messaging services including X.400 e-mail, X.500 directories, and EDI. He has spoken at numerous trade conferences and his articles appear regularly in industry publications. He is a Principal at Rapport Communication in Washington, D.C.

Mail Gatewaying

by Marshall T. Rose, Dover Beach Consulting, Inc.

[Ed.: This article is adapted from *The Internet Message—Closing the Book with Electronic Mail,* to be published by Prentice Hall. Used with permission].

Introduction

In this article we examine how different message transfer systems interact. This is through a process termed "mail gatewaying." A *mail gateway* is a special entity which sits at the boundary of two message transfer systems. Its job is to provide both connectivity and interoperability services for the user agents on both sides. We begin by looking at the theoretical basis for mail gateways. This is followed by an examination of current practices.

Theory of mail gateways

A mail gateway is more properly termed an *application-gateway,* a device which interconnects the message transfer services offered by two different protocol suites. Because of the store-and-forward nature of electronic mail, mail gateways are particularly well suited for translating between the services and contents available in the mail domains which they interconnect.

Basic principles

The basic principle for mail gatewaying lies in the observation that the envelope, headers, and body are distinct objects. In a perfect world, a mail gateway translates:

- the envelope from one mail domain to another;
- the headers from one mail domain to another; and,
- the body from one mail domain to another.

...and never allows information to commingle among these three objects.

For each kind of object, the mail gateway must be able to apply a mapping function which, in a perfect world, will be reversible without loss of information. Because mail gateways are able to access both envelope and content information, they are really neither message transfer agents nor user agents, they are somehow both simultaneously higher and lower in the message handling model. Perhaps it is this strange duality that causes so much confusion in the implementation of mail gateways.

Let's start by taking a high-level look at the three kinds of mappings, and then we'll revisit this issue with some cold pragmatism. Before we begin, let's introduce one last principle: *only* gateways should do translation; "normal" message transfer agents should never manipulate the headers or body, unless it is to add trace information to the headers. Quite a few problems have been caused by message transfer agents which violate the opaqueness principle of the message handling system.

Above all, a message transfer agent should avoid gratuitous transformations. At one site, for example, all outgoing mail is funneled through a message transfer agent which re-writes local mailbox addresses to incorporate the user's name as listed in their personnel records! Yes, both the `phrase` and `route-addr` components are re-written. Any useful information in the `phrase`, which might be placed there by a sophisticated user, is lost. Because the message transfer agent is not translating between mail domains, this behavior is wholly inappropriate.

continued on next page

Mail Gatewaying *(continued)*

Envelope mappings

The services offered by the message transfer service are largely carried in the envelope. In the Internet world, envelopes are simple: they contain an originator address, a delivery mode, and one or more recipient addresses. In contrast, in the MHS world, the envelope is much more complex, and provides (in theory) a wider range of services.

At a minimum, envelope mapping involves being able to translate between electronic mail names and addresses. To translate other services, a loss of information will almost certainly occur, e.g., there is no delivery-notification service in the Internet message transfer system.

Name and address translation is far from simple however. There are usually two parts to the process. First, a set of re-writing rules are applied to the address to map it into a canonical address form for the new mail domain. Second, the address is usually "rooted" at the mail gateway, so that any replies will go back to the mail gateway which can then reverse the re-writing. Of course, it is desirable that all mail gateways connecting two particular mail domains use the same rules. If possible, the "rooting" should be dynamic, so that the best mail gateway will be selected in the future. To understand the notion of "rooting" consider the following question:

"What is your electronic mailbox address?"

The only correct response to such a question is to first ask which mail domain should be used to provide the context for the answer. That is, a response might start with:

"In the Internet community, my mailbox address is..."

Header mappings

The headers contain four basic kinds of information:

- electronic mail addresses;
- time-stamps (dates and times);
- electronic mail message pointers; and,
- textual information.

The same address translation facilities used for the envelope must also be employed for those found in the headers. This is critical if user agent functionality such as forwarding and replying is going to work across mail domain boundaries. However, the form of translated addresses may cause problems for later message transfer agents or the recipient's user agent. So, even if a loss of information does not occur, depending on the robustness of the downstream software, anomalies or human user confusion may result.

Translation of time-stamps (e.g., the value of a `Date:` field) is probably straight-forward, so we needn't spend much time on this topic.

However, translation of message pointers (e.g., the value of a `Message-ID:` field) may be problematic unless a deterministic algorithm can be deployed. Obviously the larger the similarity between the syntaxes used in the two mail domains, or the less structured the syntax, the better.

Finally, translation of textual information (e.g., the value of a `Subject:` field) is straight-forward, unless different character repertoires are mandated.

Body mappings

In each mail domain, the body may be structured, unstructured, or pseudo-structured. If both are structured or unstructured, then mapping is fairly easy (again assuming the same character repertoires can be used). Similarly, if one is structured and the other unstructured, then mapping should also be straight-forward. The problem arises when one or both of the mail domains allow for pseudo-structured bodies. For example, in the days before MIME [3], the format used for message forwarding was pseudo-structured. This meant that the mail gateway had to apply a heuristic (i.e., non-deterministic) algorithm to decide if a message body contained a forwarded message. The problem, of course, is that in this case, the translation process is recursive: if one or more forwarded messages were encountered inside a message body, then the header and body mappings must be applied to them as well.

Imprecise mappings

It should be clear, even from the preceding high-level discussion, that the mapping functions are not going to be without information loss. All we need argue about is how much loss is acceptable to the end-user. And herein lies the rub: when the mail gateway performs its translations, it does not have any idea as to the user's wishes about what is important and what isn't.

For example, suppose a mail gateway encounters an Internet electronic mail message containing an `audio/basic` content, and this message is to be delivered to an MHS electronic mail address. As of this writing, there is no IPMS body part that corresponds to MIME's audio/basic content. What is the mail gateway to do?

- discard the content and continue processing the message; or,
- reject the message and generate an error report; or,
- attempt some translation to a different representation (in this example, this is rather doubtful).

What is the right choice? The answer is: there is no right choice. The reason is that different users have different requirements, and no matter which policy the mail gateway chooses, it is bound to choose the wrong one sooner or later (probably sooner).

Even with unstructured textual body parts, loss of information may be inevitable. For example, mappings between repertoires (e.g., NVT ASCII and EBCDIC) may cause one-way translation of character representations. Further, variations in line-termination sequences and white-space conventions may also cause problems.

For example, although the CR-LF sequence is used for interchange purposes in RFC 822, the system on which a mail gateway resides may use a different convention. Depending on the particular choice, if a body part contains a CR not followed by an LF, or an LF not preceded by a CR, then confusion may very well ensue. Similarly, on some systems, lines longer than 76 characters are wrapped or truncated; on others, lines may be padded to a particular length; on still others, lines with trailing white-space may be trimmed. Of course, none of these are conformant practices, but a message transfer agent may not have adequate control over its system environment and will simply have to make do as best it can.

Finally, all of the problems thus far have been in the realm of dealing with two mail domains. Suppose there are three. This is known as *Stefferud's Three Body Problem,* which is concisely stated as:

$$G_{2,3}(G_{1,2}(m)) \neq G_{1,3}(m)$$

continued on next page

Mail Gatewaying *(continued)*

By way of explanation, suppose one has three mail domains all interconnected. Now suppose an electronic mail message, *m*, originates from domain D_1, goes through $G_{1,2}$, traverses domain D_2, goes through $G_{2,3}$, and then is delivered in domain D_3.

Now suppose an identical message, *m'*, originates from domain D_1, goes through $G_{1,3}$, and then is delivered in domain D_3. Stefferud claims (and the author agrees) that the nature of imprecise mappings means that upon final delivery, *m* and *m'* will have entirely different semantics in the header and probably even body.

Why is this a problem? The message transfer agents operating within a mail domain may select different mail gateways, depending on availability, cost, and so on. If *m* and *m'* have suffered a different set of transformations, then user agent functionality will greatly suffer.

For interested readers

Defining the set of mappings between two mail domains is difficult, tedious, and generally a dirty business. For example, the mappings between Internet mail and MHS are defined in [1]. To give the reader an idea of the complexity of such mappings, this document is over 110 pages long and discusses only envelope and header mappings! A supplemental work effort is just beginning in the IETF to define mappings between message bodies for the two mail domains.

Mail gateways in practice

In practice, mail gatewaying is an absolute mess. We'll look at five examples of mail gateways which are broken in various ways and then try to identify the underlying causes.

Example 1: Envelope fixation

Suppose we have an SMTP envelope that contains several addresses for the same, foreign mail domain. An SMTP connection is established to the mail gateway which interconnects the Internet message transfer service to this other mail domain. Now let's say that some of the addresses given in the RCPT commands are rejected, but others are not. The mail gateway in question will refuse the DATA command which follows. That is, if it can't deliver to all of the recipient addresses, then it refuses to delivery to any of them.

If this gateway does accept the electronic mail message, it keeps the `From:` and `Subject:` headers, and discards the rest of the message headers. It then constructs a new message header consisting of:

- a translated `From:` header;
- the old `Subject:` header;
- a `Date:` header set to the time when the mail gateway received the message;
- a `To:` header consisting of the addresses given as recipients during the SMTP transaction.

Of course, since the headers of the electronic mail message may have contained addresses from other mail domains, this information is lost, and the user agent's reply functionality does not work as it should.

Example 2: Header examination

For a second example, suppose we have an RFC 822 message. The minimal set of headers present is:

```
Date:

From:
```

and either a `To:` header or a `Bcc:` header.

Note that no recipient addresses need be present. For example, according to RFC 822:

- the `Bcc:` header needn't contain any addresses; and,
- the `To:` header could have an address group which contains no addresses, e.g., `To: Reviewers:;` is perfectly valid.

One of the alleged selling points of MHS is that it is supposed to have been developed for commercial providers of electronic mail services. And, as we all know, commercial providers take these things seriously and offer robust, production services. So, let's see what happens when we take an electronic mail message with these minimal headers and try to pass it through an 822/MHS gateway operated by a commercial service provider. In one case, the message was accepted for delivery and then silently discarded. In fact, there wasn't even a log entry indicating that the message had been accepted. In a case dealing with a different service provider, an error report was returned, in which the message body consisted solely of the original message, and buried in the message headers was the "reason" for the problem:

```
Not-Delivered-To:!anybody due to 10 Invalid Parameters
        Message header has unacceptable format
```

Now isn't that informative? It's a good thing that the recipient of this error report wasn't using a header filter when displaying this message. Otherwise, he never would have even seen this extremely helpful diagnostic.

The next part of the story is somewhat amusing. Someone started sending electronic mail to the PostMaster for the mail gateway asking for an explanation. Over the next few months, several messages went unanswered, until one day, a reply was received, which appeared to have been automatically generated. Paraphrasing, the text of the reply went something like this:

> *Thank-you for your message. It was been entered into our trouble-ticket tracking system as report #1729.*

Further investigation, to the author's knowledge, never occurred, and the problem remains to this day. Perhaps the last line of the reply was missing, e.g.,

> *Now serving #3.*

Example 3: Uniformity of translation

For another example, the author regularly receives messages from a client whose mailbox is in a different mail domain. The mail gateway between the two domains has an interesting property: it will translate the addresses in the `From:` and `To:` headers, but not the `cc:` headers. As a result, when the author wants to reply to such a message, the reply will reach only the author of the original message, the `cc:` recipients are unreachable.

Example 4: Header mapping

For the fourth example, there are some mail domains which support only a subset of headers found in an Internet message. When a message crosses into this domain, the gateway takes any headers it doesn't recognize and appends them to the body of the message. As a result, if the body contains any kind of structured object, then it is likely to be corrupted. In this example, the gateway considers preserving the body of the message to be less important than losing the information contained in the headers.

continued on next page

Mail Gatewaying *(continued)*

Of course, such behavior assumes that a human user will be making use of the body, rather than some kind of program, as these trailing headers won't be directly interpreted by the user agent.

Example 5: Message body translation

As the final example, there are some enclaves in the Internet which, for historical reasons, write the domain name part of an electronic mail address in reverse order, e.g.,

`local@us.ca.mtview.dbc` instead of `local@dbc.mtview.ca.us`

Their mail gateways perform the necessary reversal. Alas, these gateways do not check to see if the message body contains forwarded messages. As a result, when an electronic mail message passes through the mail gateway and is subsequently burst, the interior messages have unusable addresses.

The common theme

The common theme is that these examples all result in the same problem: user agent functionality is compromised. The underlying cause is that the mail gateways in question do not honor the boundaries between envelope, headers, and body:

- in the first example, the mail gateway throws away information in the headers, only to (incorrectly) regenerate it later on from information in the envelope;
- in the second example, the mail gateway is likely scrutinizing the headers looking for addresses that it might service, instead of looking in the envelope where the recipient addresses truly reside;
- in the third example, the mail gateway should be examining all the headers, but looks instead at an incorrect subset;
- in the fourth example, the mail gateway copies some headers to the body, in order to avoid losing information; and,
- in the fifth example, the mail gateway should look in the body of the electronic mail message, but doesn't.

What can be done? Some argue that we need to connect the various mail domains using a single mail transit backbone which implements a superset technology. The author disagrees. Using a superset technology doesn't guarantee mappings which are free of information loss. Keep in mind that although entry into the superset backbone may not result in a loss of information, exit from the superset backbone certainly might, particularly if there isn't an obvious service mapping between the originating and destination mail domains.This suggests the only practical solution is one in which:

- only minimal services are supported across the backbone; and,
- gateway "rooting" occurs dynamically.

This may seem counter-intuitive, but the author suspects that the solution to our problems lies in choosing the correct subset technology for an interconnected mesh, rather than choosing a superset technology for a ubiquitous backbone.

References

[1] Kille, Stephen, E., "Mapping between X.400(1988) and RFC 822," RFC 1327, May 1992.

[2] Rose, M. T., *The Internet Message—Closing the Book with Electronic Mail,* ISBN 0-13-092941-7 (book to be published by Prentice Hall, October 1992.)

[3] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1341, June 1992.

[4] Moore, K., "Representation of Non-ASCII Text in Internet Message Headers," RFC 1342, June 1992.

[5] Borenstein, N., "A User Agent Configuration Mechanism For Multimedia Mail Format Information," RFC 1343, June 1992.

[6] Borenstein, N., "Implications of MIME for Internet Mail Gateways," RFC 1344, June 1992.

[7] Simonsen, K. "Character Mnemonics & Character Sets," RFC 1345, June 1992.

[8] Crocker, D., "Standard for the format of ARPA Internet Text Messages," RFC 822, August 1982.

[9] CCITT, "Data Communications Networks Message Handling Systems Recommendations X.400–X.430," 1984.

[10] CCITT/ISO, "Data Communications Networks Message Handling Systems Recommendations X.400–X.420," aka ISO 10021, 1988.

MARSHALL T. ROSE is Principal at Dover Beach Consulting, Inc., a California-based computer-communications consultancy. He spends half of his time working with clients, and the other half involved in self-supported, openly-available projects. Rose lives with internetworking technologies, such as TCP/IP, OSI, network management, and directory services, as a theorist, implementor, and agent provocateur. He is the author of three professional texts—on Open Systems Interconnection (*The Open Book*), Internet Management (*The Simple Book*) and OSI Directory Services (*The Little Black Book*)—all published by Prentice-Hall. His fourth and final book, *The Internet Message,* on electronic mail, will be published this fall. Rose received the Ph.D. degree in Information and Computer Science from the University of California, Irvine, in 1984. His subscriptions to *The Atlantic* and *Rolling Stone Magazine* are in good standing. He can be reached on the Internet as: `mrose@dbc.mtview.ca.us`.

Write to *ConneXions*!

Have a question about your subscription? Are you moving, and need to give us your new address? Suggestions for topics? Want to write an article? A letter to the Editor? Have a question for an author? Need a *ConneXions* binder? Want to enquire about back issues? (there are now sixty-six to choose from; ask for our free 1987–1992 index booklet). We want to hear from you. Contact us at:

ConneXions—The Interoperability Report
480 San Antonio Road, Suite 100
Mountain View, CA 94040–1219
USA
Phone: +1 415-941-3399 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-949-1779
E-mail: `connexions@interop.com`

Book Reviews

Electronic Mail: An Introduction to the X.400 Message Handling Standards, by Sara Radicati, McGraw-Hill, Inc., ISBN 0-07-051104-7, 1992.

Electronic Mail (EM) is a text with several faces, some good, some bad. *EM* is a welcome addition to the small collection of texts focused on electronic mail and as such will be a valuable tool for those just learning about electronic mail or the more seasoned person looking for specifics on the OSI standards for electronic mail, X.400.

Credentials

EM's author, Sara Radicati, has impeccable credentials for presenting the OSI standards for electronic mail and directory. She has been intimately involved in many aspects of their development, and I looked forward to reading the text. *EM* left this reviewer with a clear set of likes and dislikes regarding the text. *EM* will be a good book for some and others will, no doubt, find it disappointing. *EM* is a book about X.400, the OSI standard for electronic mail.

The bad parts

There is a lot to be said for picking a good title when publishing a technical book. In the case of this reviewer the title, "Electronic Mail" set an expectation for a general text on electronic mail. However, EM—as its subtitle, "An Introduction to the X.400 Message Handling Standards," suggests—is focused on the OSI standards for electronic messaging, X.400. This is unfortunate and not immediately obvious to the reader. In fact, this reviewer's biggest complaint about *EM* is just that, the text initially presents itself as a generic piece on electronic mail but then focuses and provides its highest quality material on the topic of X.400. Early sections in the text do deal, somewhat inaccurately, with the Internet mail protocols, SMTP and RFC 822.

Winding up the bad news on *EM,* the text takes a dubious position regarding the relative installed bases of Internet electronic mail and OSI's X.400 technology. *EM* positions them as peers in the installed base and then continues based on this erroneous basis. Further, no acknowledgment of the many other (non Internet and non OSI) electronic mail communities such as Novell's *MHS* and Lotus's *cc:Mail* is made. This is a significant oversight as it is virtually impossible to place either Internet or OSI technology in perspective without an appreciation for the fact that many other sizable communities do exist.

The good parts

EM has a number of chapters on X.400 that provide a good reference on the topic. With the exception of someone who is directly implementing an X.400 application, *EM* provides a very concise way to read about X.400 vs. plowing through the CCITT and ISO standards documents—a welcome relief. *EM*'s second chapter provides a great summary of store-and-forward electronic mail principles. This material will prove to be particularly valuable for those trying to come up to speed on electronic mail concepts. Something those of us steeped in the technology often take for granted. For the trivia buff this section has some little known facts about the early days of Internet mail and its development.

As the text begins to wrap up, the features of X.400 security are presented. Again, a consistently thorough summary of the standards and some implementation profile work is presented. Notably lacking is any mention of competing security schemes, such as PEM in the Internet, that presents substantial competition for X.400 security features.

In the final chapter the text presents a variety of very interesting topics such as gateways from existing mail communities to X.400, standardized programming APIs and work in progress with the standards organizations.

Conclusion

In conclusion, this reviewer is "middle of the road" on the book *Electronic Mail.* For a specific audience, those wanting to learn about the X.400 standards, this book will be of great value. Unfortunately, for the reader not focused on X.400, the text falls short of providing enough valuable information outside the realm of X.400.

—Chris Moore, General Magic

Computing Across America: The Bicycle Odyssey of a High-Tech Nomad, by Steven K. Roberts, Learned Information, Inc. (Medford, NJ: 1988), 347 pp., ISBN 0-938734-18-0.

Normally, we don't review books that are four years old, but the marketing on the book *Computing Across America* was so bad that we're only just now learning of its existence. The author of this unusual book is Steve Roberts, aka "The Guy with the Bike."

The bike

You know, the bike you read about in *USA Today* and saw at INTEROP 92 Spring or on *The Phil Donahue Show.* That's right, the 580-pound, multiprocessing, multi-functional contraption that contains more computer equipment than your average Novell dealer. (For a detailed description of the bike, see "The Guy With the Bike," in *ConneXions,* Volume 6, No. 8, August 1992).

Ever wonder how something this evolved ever got started? *Computing Across America* is Steve Robert's narrative of how he quit his life in suburban Ohio, sold everything and hit the road. A change in lifestyle is certainly nice every now and again, but why, you might ask, a recumbent bicycle with a head mouse?

In 1983, in a moment of suburban mid-life angst, Steve Roberts asked himself what he liked most in life. Bicycles, computers, and women led the list. Mowing the lawn and sitting in cubicles weren't on the list.

Evolution

The natural answer was to get a recumbent bicycle, buy one of the new laptops, and pedal 10,000 miles looking for coeds. In this fascinating book, we see Roberts go through an evolution, learning how to lead his life the way he wants.

Technically, Roberts also evolves, taking the concept of the laptop computer on a bicycle and extending it to a solar-powered Winnebiko, an example of engineering at its best. Today, the Winnebiko has undergone another metamorphosis and has become the BEHEMOTH (*Big Electronic Human-Energized Machine...Only Too Heavy).*

Big hit

Computing Across America tells you how this manic pedal person got started. Roberts is now hard at work on a new book, to be published by Doubleday and if the humorous and elegant writing style of *Computing Across America* holds, the new book will be a big hit.

—Carl Malamud

Extending the reach of e-mail

by Ole J. Jacobsen, Interop Company

Introduction

Pagers (sometimes called "beepers") have of course been around for a long time. For INTEROP 91 Fall, I got a simple numeric pager which was intended to be used as a way of summoning me to one specific area of the INTEROP campus. Since I also carry a portable cellular phone, the pager wasn't used a great deal, but I decided to keep it anyway and started experimenting with various "notification services."

Notification

For instance, if you call my office number and leave voice mail, the voice mail system will notify me within a couple of minutes by dialing the pager and dumping the simple numeric string "2515" which corresponds to my phone extension.

Also, I have voice mail on my cellular phone which will "kick in" if the phone is in use, out of the service area, or turned off (a rare occurrence). The cellular voice mail system will also page me with (a different) numeric string to indicate that someone left a message. This is particularly useful since you get charged "airtime" for checking to see if you have any messages, even if you don't have any—and even if you're using a "landline" phone to do so.

Finally, I recently bought an all-digital answering machine which includes the ability to "transfer" messages. Normally, the machine would call a preprogrammed number and play back a recorded message such as "Please bring Ole to the telephone, I have a message for him." I would then enter my security code and take control of the machine to play back messages. As you might already have guessed, the machine is now set up to call my pager and leave a "message" of recorded touch tones. This notifies me when there is a message on my answering machine at home.

RadioMail

So, what next? Well, recently I upgraded the pager to an alpha-numeric model and subscribed to a service called *RadioMail* from Anterior Technology (and of course alpha-numeric paging service from Pac*Tel Paging). Nowadays, if you send me electronic mail, the first 120 or so characters of the message will be displayed on the pager. (I used to get about 450 characters, but the system has become so popular that Pac*Tel now restricts the message size to prevent system overload). It's not two-way, but it sure has changed my e-mail habits. Instead of logging in several times a day and wading through all the messages—looking for those that need my attention—I can now read e-mail when I need to.

The system is of course not limited to local service, if you need nationwide *RadioMail* coverage, all you need to do is subscribe to a nationwide paging system, such as that offered by *SkyTel*.

"My e-mail goes wherever I go."

Call for Papers

CONVENTION UNIX 93 will be held from March 22 to 26, 1993, at CNIT, Paris-La-Défense. The convention is sponsored by AFUU (The French UNIX Users Association).

Themes

Today's users have increasingly sophisticated operational needs. They want open and advanced solutions which make the full spectrum of possibilities available to them. CONVENTION UNIX 93 promises to clarify ánd demonstrate the concepts of interoperability and multimedia, from both a technical and economical standpoint, to answer the following questions:

- What functionalities do interoperability and multimedia provide users and businesses?
- What advantages do they offer?
- What are the economic implications?
- What concrete, demonstrable examples exist today?

Topics

The Program Committee is interested in both technical papers and syntheses of different approaches. Papers can deal with users experiences, industry and development strategies, technical innovation from the research world or an overview of principles in a particular area. Papers that address the following topics are invited:

- *Applications and Interfaces:* implementation of standards, ergonomics, collection and compression of video images, image synthesis, animation, scene composition, assisted vision, speech/voice recognition, video or teleconferencing, virtual reality, ...
- *Tools and Basic services:* distributed systems, object oriented databases, communication protocols, toolkits, RPC, ...

Tutorials

The tutorial program gives users an opportunity to increase their knowledge and find solutions to specific problems through lectures on precisely defined topics. The objective is to present the state of the art in interoperability and multimedia implementation. People interested in presenting a tutorial are invited to contact Hugues Leroy (`Hugues.Leroy@irisa.fr`), the chairman of the program committee as soon as possible.

Submissions

Authors are encouraged to submit the final version of their papers as quickly as possible. Proposals must have a title, name and affiliation of the author, and the complete text of the paper (5 to 10 pages) or, at the very least, an extended abstract (2 pages). Send your proposal to:

A.F.U.U.
Secretariat de CONVENTION UNIX 93
11, rue Carnot
94270 Le Kremlin-Bicetre
FRANCE
Phone: +33 1 46 70 95 90
Fax: +33 1 46 58 94 20
E-mail: `afuuconf@irisa.fr`

Details about submission formats can be obtained from the above address.

Important dates

Full Papers of extended abstracts due:	October 15, 1992
Notification to authors:	October 30, 1992
Final papers due:	December 18, 1992

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1